

VŠB – Technická univerzita Ostrava  
Fakulta strojní  
Katedra automatizační techniky a řízení

**Systém monitorování a řízení technologie**  
**A System for Monitoring and Control of a  
Technological Unit**

Student: Jaromír Zavadil  
Vedoucí diplomové práce: Doc. Ing. Lenka Landryová, CSc.

Ostrava: 22. 5. 2009

### Prohlášení diplomanta

Prohlašuji, že jsem celou diplomovou práci včetně příloh vypracoval samostatně pod vedením vedoucího diplomové práce a uvedl jsem všechny použité podklady a literaturu.

V Ostravě dne .....

.....  
Jaromír Zavadil

Prohlašuji, že

- byl jsem seznámen s tím, že na moji diplomovou práci se plně vztahuje zákon č.121/2000 Sb. – autorský zákon, zejména §35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a §60 – školní dílo.
- beru na vědomí, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen VŠB-TUO) má právo nevýdělečně ke své vnitřní potřebě diplomovou práci užít (§35 odst. 3).
- souhlasím s tím, že jeden výtisk diplomové práce bude uložen v Ústřední knihovně VŠB-TUO k prezenčnímu nahlédnutí a jeden výtisk bude uložen u vedoucího diplomové práce. Souhlasím s tím, že údaje o bakalářské práci budou zveřejněny v informačním systému VŠB-TUO.
- bylo sjednáno, že s VŠB-TUO v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu §12 odst. 4 autorského zákona.
- bylo sjednáno, že užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití mohu jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).
- beru na vědomí, že odevzdáním své práce, souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, bez ohledu na výsledek její obhajoby.

V Ostravě dne .....

.....  
Jaromír Zavadil

## ANOTACE DIPLOMOVÉ PRÁCE

Zavadil, J. *Systém monitorování a řízení technologie*. Ostrava: katedra automatizační techniky a řízení, Fakulta strojní VŠB-Technická univerzita Ostrava, 2009. 57 s. Diplomová práce, vedoucí: Landryová, L.

Diplomová práce se zabývá problematikou systémů určených k monitorování a řízení technologie. Konkrétně monitorováním a řízením elektrických pohonů. První část je věnována úvodu k problematice asynchronních motorů a způsobům jejich řízení. Následuje popis reálného modelu umístěného ve školní laboratoři. Cílem práce bylo sestavit, zapojit a zdokumentovat úlohu řízení a monitorování asynchronního motoru pomocí frekvenčního měniče a nadřazeného systému (PLC). Pozornost je věnována popisu jednotlivých komponent, jak hardwaru, tak i softwaru. Jejich použití a konfiguraci. Další část práce se zabývá tvorbou programů pro PLC a následnou vizualizaci. Výsledný systém představuje využití těchto prostředků v praxi a měl by sloužit pro výuku studentů.

## ANNOTATION OF THESIS

Zavadil, J. *A System for Monitoring and Control of a Technological Unit*. Ostrava: Department of Control Systems and Instrumentation, Faculty of Mechanical Engineering VŠB-Technical University of Ostrava, 2009. 57 p. Thesis, head: Landryová, L.

Diploma thesis deals with a system for monitoring and control of technologies, specifically monitoring and control of electro drives. The first part is introducing the asynchronous drives and methods for their control. Furthermore, a real model from a department laboratory is described. A goal of this work was to build and configure system architecture and to document a task for control and monitoring of asynchronous drive by frequency converter and a superior system (PLC). The focus is placed on each component description, hardware and software, their use and configuration. Programs created for PLC and for visualization are also described. Designed system represents an application of these software and hardware means in industrial automation and will help students during their classes in a laboratory.

## Obsah diplomové práce

Seznam použitých zkratk .....	6
1 Úvod .....	7
2 Asynchronní motory a jejich řízení .....	8
3 Zařízení vybavené frekvenčním měničem ACS 800 .....	11
3.1 Asynchronní motor .....	12
3.2 Frekvenční měnič .....	14
4 Zapojení s PLC .....	18
4.1 PLC .....	20
4.2 Nastavení měniče .....	22
5 Použitý software .....	26
5.1 PLC Control Builder AC 800M .....	27
5.2 OPC Server for AC 800M 4.1 .....	28
5.3 SoftController 4.1 .....	29
5.4 DriveWindow 2.12 .....	29
5.5 InTouch 10.0 .....	31
5.6 Wonderware Application Server 3.0 .....	33
5.7 OPCLink 8.0 .....	37
5.8 FactorySuite Gateway .....	37
6 Demonstrační úloha a programovací prostředí .....	39
6.1 Řídicí program pro PLC .....	43
6.2 Aplikace pro vizualizaci v InTouch .....	44
6.3 Nastavení komunikace .....	48
7 Návrh směru dalšího řešení .....	51
8 Závěr .....	52
Použitá literatura .....	54

## Seznam použitých zkratk

DDE	Dynamic Data Exchange – dynamická výměna dat – protokol MS Windows pro výměnu dat mezi aplikacemi
DOS	Disc Operating System – diskový operační systém
DTC	Direct Torque Control – přímé řízení momentu
FBD	Function Block Diagram – funkční blokové schéma
HMI	Human Machine Interface – rozhraní člověk-stroj
IDE	Integrated Development Environment – integrované vývojové prostředí
IAS	Industrial Application Server – průmyslový aplikační server
IL	Instruction List – instrukční list
IP	Internet Protocol – standardní síťový protokol
LD	Ladder Diagram – kontaktní schéma
MES	Manufacturing Execution Systems – systém pro řízení a analýzu chodu a výkonnosti výroby
MMS	Manufacturing Message Specification – specifikace výrobních zpráv – Protokol pro uživatelsky orientovanou komunikaci aplikačních programů s HW výrobního procesu
OLE	Object Linking and Embedding – vkládání a propojování objektů – Protokol MS Windows
OPC	OLE for Process Control – rozhraní pro komunikaci mezi rozdílnými aplikacemi, zařízeními atd.
PC	Personal Computer – osobní počítač
PCMCIA	Personal Computer Memory Cards International Association
PLC	Programmable Logic Controller – programovatelný logický automat
SCADA	Supervisory Control and Data Acquisition – supervizní řízení a sběr dat
SFC	Sequential Function Chart – sekvenční funkční diagram
SMC	System Management Console – nástroj pro správu systému ArchastrA
ST	Structured Text – strukturovaný text
VMS	Virtual Memory System – systém virtuální paměti

# 1 Úvod

Cílem projektu je seznámení s vybranou technologií určenou k monitorování, způsoby a systémy monitorování, zejména pak s nástroji a prostředky, které tento proces umožňují.

Nejprve se věnuji úvodu do problematiky asynchronních motorů. Jejich rozdělení, princip činnosti a způsoby jejich řízení. Dále následuje popis vybraného technologického procesu (zařízení). Jedná se o demonstrační úlohu, která se skládá z asynchronního motoru a frekvenčního měniče. Tuto sestavu na VŠB-TUO dodala firma ABB.

Po seznámení s tímto zařízením uvádím popis PLC, které jsem k úloze následně připojil jako nadřazený systém řízení. Popisuji způsob propojení jednotlivých prvků včetně jejich nastavení.

Další kapitola se věnuje použitému softwaru, určenému k obsluze jednotlivých komponent tohoto zařízení nebo jejich programování. Dále pak následuje popis softwaru použitého pro vizualizaci procesu. Jedna z částí této práce popisuje programovací prostředí PLC Control Builderu a demonstrační úlohu dodanou k sestavě. Je popsáno jak úlohu spustit, nahrát do PLC a vyzkoušet její funkce. Následně uvádím popis vlastní řídicí aplikace pro PLC, kterou jsem za účelem následné vizualizace navrhnul. Dále pak popisují tvorbu vizualizační aplikace v programu InTouch za pomoci nového vývojového prostředí ArchestrA IDE. Pozornost je také věnována komunikaci jednotlivých částí úlohy. Nastavení komunikace je závěrečným oddílem tohoto projektu.

## 2 Asynchronní motory a jejich řízení

Asynchronní motory jsou nejrozšířenější pohony v elektrotechnice. Jedná se o konstrukčně nejjednodušší, a proto nejlevnější elektrické motory s vysokou spolehlivostí. Napájecí napětí může být buď jednofázové, nebo trojfázové (používanější). Nevýhodou těchto zařízení je pouze velký proudový náraz při rozběhu.

Konstrukce asynchronního motoru se skládá ze dvou hlavních částí – statoru (pevná část) a rotoru (pohyblivá část, označována také jako kotva). **Stator** je u většiny typů motorů prakticky stejný. Skládá se z nosné kostry motoru, statorových plechů a statorového vinutí. **Rotor** je hřídel s nalisovanými rotorovými plechy s drážkami, které tvoří druhou část magnetického pole.

Podle provedení rotoru se asynchronní motory dělí na:

- **motory s kotvou nakrátko** - v drážkách rotoru jsou nalisovány neizolované měděné, mosazné nebo hliníkové tyče, které jsou na obou koncích spojeny zkratovacími kroužky. Tyče spolu s kroužky mají podobu klece, proto se tomuto typu rotoru říká také klecový rotor.
- **motory s kotvou kroužkovou** - na hřídeli jsou kromě svazku rotorových plechů i sběrné kroužky. V drážkách plechů je uloženo trojfázové vinutí rotoru z izolovaných vodičů, které je zapojeno většinou do hvězdy. Na tři sběrací kroužky je připojeno vinutí rotoru, ke kterým lze připojit činné odpory sloužící k rozběhu motoru.

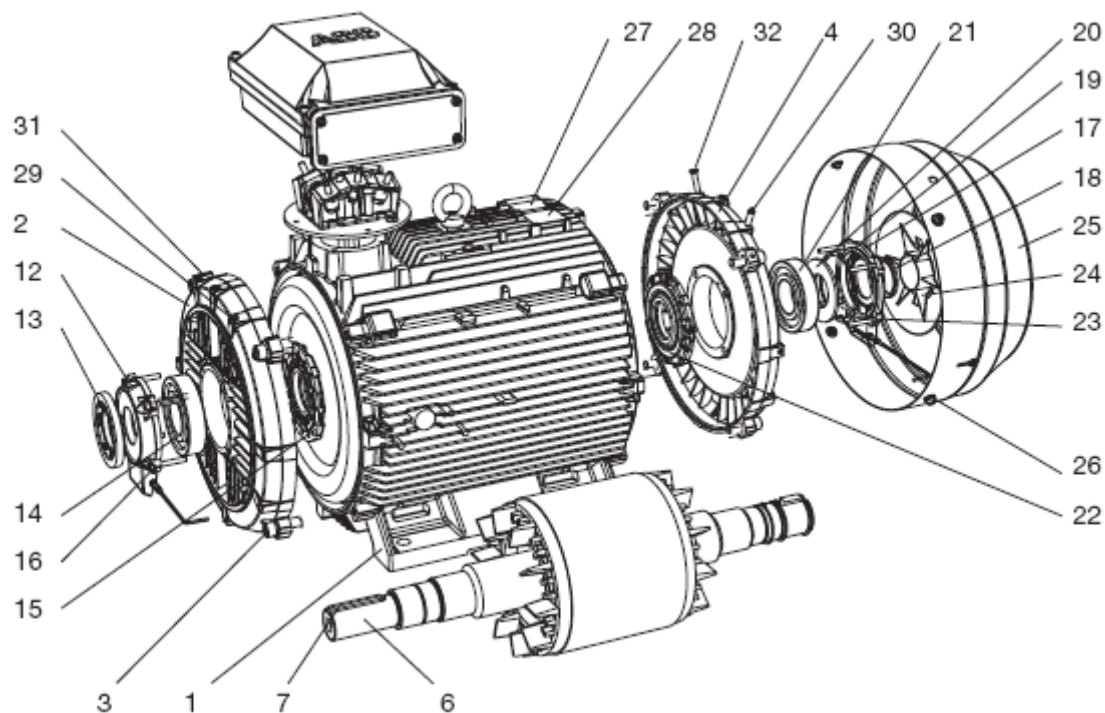
Základem činnosti asynchronního motoru je vytvoření točivého magnetického pole, které vznikne průchodem střídavého trojfázového proudu vinutím statoru. Toto magnetické pole indukuje v rotoru napětí a vzniklý proud vyvolává sílu otáčející rotorem.

Otáčky točivého pole jsou dány kmitočtem napětí odebíraného ze sítě a počtem pólů trojfázového motoru. Rotor se nikdy nemůže otáčet stejnými otáčkami jako magnetické pole statoru. Pokud by se tak stalo, pak by se rotor a magnetické pole vůči sobě nepohybovaly a tím by se ani neindukovalo napětí a nevznikala by točivá síla. Míra rozdílu otáček magnetického pole a rotoru je nazývána skluz, udávána v procentech.

Při spouštění asynchronního motoru je záběrový proud až 7 krát vyšší než hodnota nominálního proudu. Tím vznikají v síti velké proudové rázy při poměrně malém



záběrovém momentu. Proto je přímé spouštění povoleno pouze pro motory s výkonem přibližně do 3 kW.



**Obr. 1** Typická sestava třífázového motoru velikosti 315 od ABB [ABB 2006]

- |                                   |                                   |
|-----------------------------------|-----------------------------------|
| 1 Rám statoru                     | 17 Vnější kryt ložiska, N-end     |
| 2 Víko, D-end                     | 18 Těsnění, N-end                 |
| 3 Šrouby na víko, D-end           | 19 Vlnová pružina                 |
| 4 Víko, N-end                     | 20 Kroužek, N-end                 |
| 5 Šrouby na víko, N-end           | 21 Ložisko, N-end                 |
| 6 Rotor s hřídelí                 | 22 Vnitřní kryt ložiska, N-end    |
| 7 Pero hřídele, D-end             | 23 Šrouby pro kryt ložiska, N-end |
| 8 Kryt svorkovnice                | 24 Ventilátor                     |
| 9 Svorkovnice                     | 25 Kryt ventilátoru               |
| 10 –                              | 26 Šrouby pro kryt ventilátoru    |
| 11 Šrouby pro kryt svorkovnice    | 27 Výkonnostní štítek             |
| 12 Vnější kryt ložiska, D-end     | 28 Mazací štítek                  |
| 13 Těsnění, D-end                 | 29 Maznice, D-end                 |
| 14 Ložisko, D-end                 | 30 Maznice, N-end                 |
| 15 Vnitřní kryt ložiska, D-end    | 31 SMP maznice, D-end             |
| 16 Šrouby pro kryt ložiska, D-end | 32 SMP maznice, N-end             |

Regulaci otáček těchto motorů lze provádět třemi způsoby:

- Regulace pomocí skluzu – pouze pro motory s kroužkovou kotvou
- Regulace změnou kmitočtu – používá se u motorů s kotvou na krátko
- Regulace změnou počtu pólů – lze dosáhnout pouze skokové změny otáček

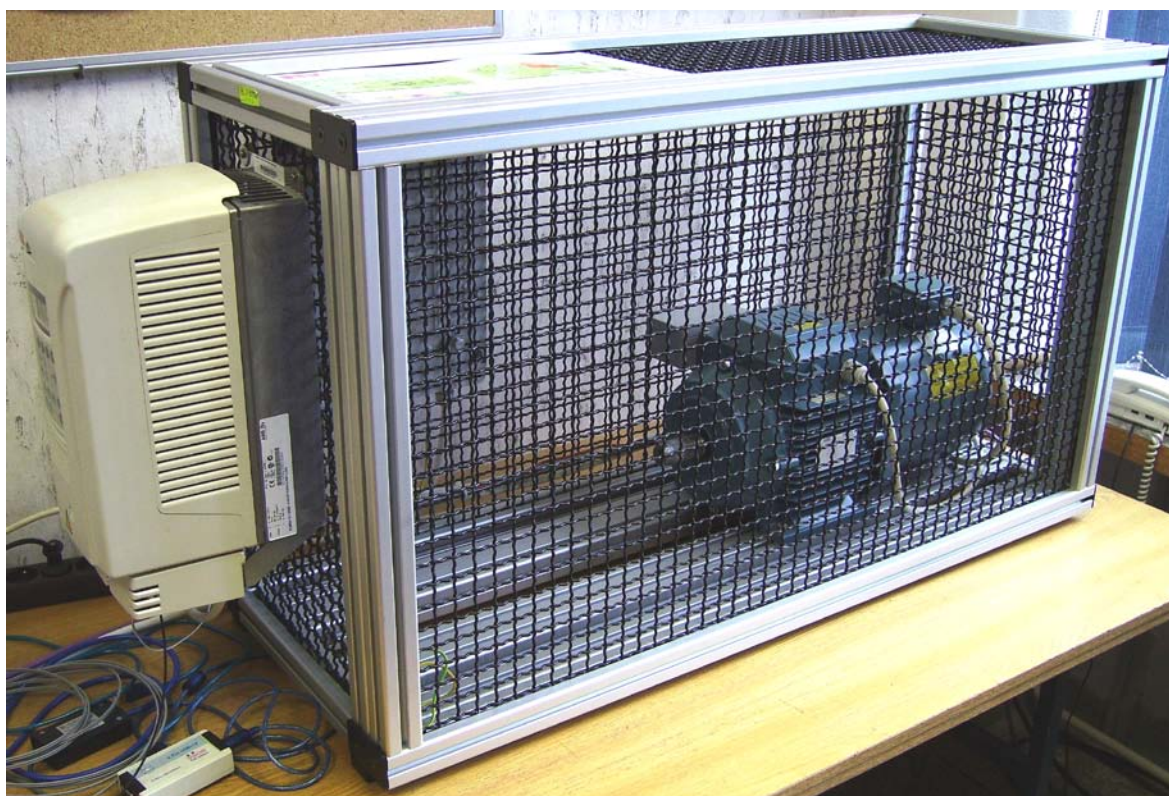
### **Regulace otáček změnou kmitočtu**

Motor použitý v rámci této práce využívá regulaci otáček pomocí změny kmitočtu. Připojením frekvenčního měniče můžeme řídit napětí a tím i vytvářené magnetické pole statoru. Frekvenční měnič, který je součástí tohoto projektu, umožňuje dva způsoby řízení:

- ***Skalární řízení*** – jeden z prvních způsobů řízení frekvenčních měničů. Pracuje na principu lineární změny napětí motoru a frekvence do jmenovitých otáček motoru.
- ***Přímé řízení momentu (DTC)*** – využívá se zejména v průmyslových aplikacích. Tento způsob řízení je schopen zajistit plný moment motoru od nulových otáček až po jmenovité otáčky. Což jiné druhy řízení neumožňují. Další výhodou je možnost přímého zadávání reference momentu. To bylo dříve možné pouze u stejnosměrných motorů

### 3 Zařízení vybavené frekvenčním měničem ACS 800

Demonstrační zařízení, které bude v rámci této práce monitorováno, se skládá z frekvenčního měniče ACS 800 a asynchronního motoru M2AA 112M(Obr. 2). Sestavu dodala firma ABB. Vše je umístěno ve školní laboratoři F204. V této kapitole představím a rozeberu postupně jak motor, tak i měnič. K sestavě je dále připojen také stolní počítač vybavený příslušným softwarem. Toto PC slouží ke „vzdálenému“ ovládání měniče, zejména pak ke konfiguraci parametrů jeho firmware. Propojení měniče s PC je realizováno pomocí PCMCIA karty NDPC-12 a rozhraní optického ModuleBus. Jelikož PC použité v našem případě není vybaveno slotem pro karty PCMCIA, musel být pro připojení použit ještě adaptér PCI/PCMCIA, který umožnil připojení PCMCIA karty do slotu PCI. Ke kartě PCMCIA je firmou ABB dodáván obslužný software DriveWindow, pomocí něj lze měnič snadno konfigurovat a ovládat. Popis použitého softwaru, tedy i DriveWindow, uvádím v kapitole č. 5.



Obr. 2 Demonstrační zařízení vybavené frekvenčním měničem

### 3.1 Asynchronní motor

Motor použitý v tomto projektu je umístěn v ochranné kleci, na které je upevněn frekvenční měnič a také napájecí skříň s pojistkami. Jedná se o nízko výkonový hliníkový asynchronní motor s kotvou na krátko ovládaný přes frekvenční měnič ACS 800 (ten popisují v další podkapitole). Typ motoru je M2AA, velikost 112M.

Z důvodu předpokládané vyšší zátěže při různých testovacích režimech je motor doplněn přidaným větracím systémem od firmy Kurt-Maier. Pro monitorování teploty motoru je motor vybaven termistorovým ochranným relé. Jedná se o termistor PTC. Tyto termistory mají při nízkých teplotách malý odpor, který od určité hodnoty teploty prudce roste. Pro přesné měření otáček je k motoru připojen inkrementální snímač otáček (viz Obr. 3). V současnosti k motoru není připojen žádný další systém (zátěž).



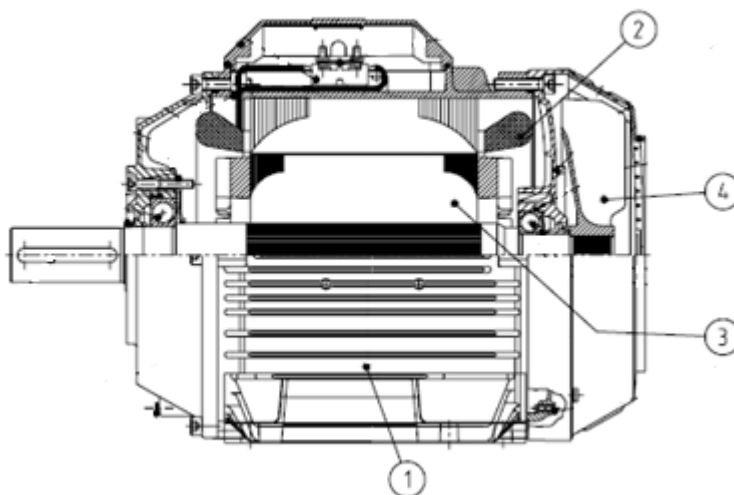
Obr. 3 Inkrementální snímač otáček

Parametry motoru jsou následující:

Typ motoru: M2AA 112M

Kód produktu: 3GAA 111 001-ADA

Parametry:	Jmenovité napětí motoru	400V
	Jmenovitý proud motoru	7,2A
	Jmenovitá frekvence	50Hz
	Jmenovité otáčky	2850 ot./min.
	Jmenovitý výkon	4kW
	Účinník $\cos\varphi$	0,91
	IP	55



**Obr. 4 Motor M2AA 112M [ABB – upraveno]**

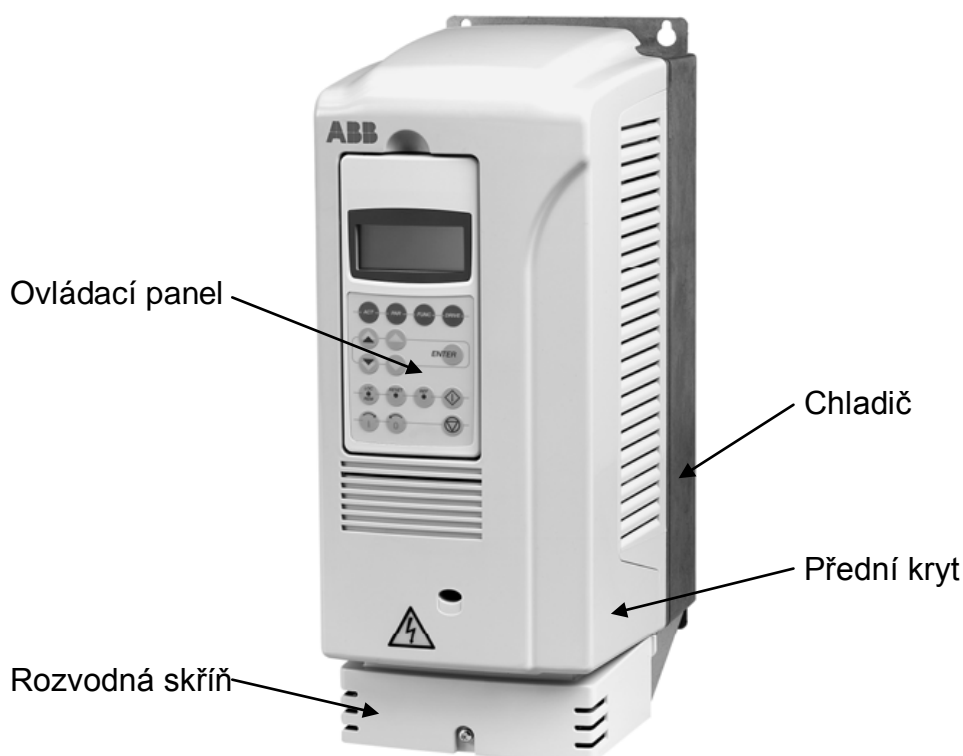
**1. statorová mřížka, 2. vinutí statoru, 3. rotor s krytem, 4. větrák**

### 3.2 Frekvenční měnič

Demonstrační zařízení je vybaveno frekvenčním měničem řady ACS 800. Tyto měniče jsou určeny pro řízení střídavých motorů a disponují dvěma způsoby řízení:

- Skalární řízení
- DTC (Direct Torque Control) – přímé řízení momentu

U měničů ACS 800 se používá hlavně řízení DTC, které umožňuje přesně regulovat otáčky motoru a jeho točivý moment i bez zpětné vazby. Skalární řízení je doporučeno používat pouze v případech, kdy není možné použít řízení DTC (např. více motorové aplikace s různými typy motorů atd.). Podrobněji se způsobům řízení střídavých motorů věnuje kapitola č. 2.



Obr. 5 Frekvenční měnič ACS 800

Měnič nabízí dvě možnosti ovládání a nastavení. Základní nastavení umožňuje odnímatelný panel CDP 312R. Další možností je ovládání měniče vzdáleně externími zařízeními prostřednictvím komunikační sítě. K tzv. „vzdálenému“ ovládání slouží například softwarové prostředí DriveWindow. Programovému vybavení a tedy i programu DriveWindow se věnuji v kapitole č.5. Z tohoto důvodu nyní rozeberu jen ovládání pomocí panelu a princip komunikace vzdáleného ovládání.

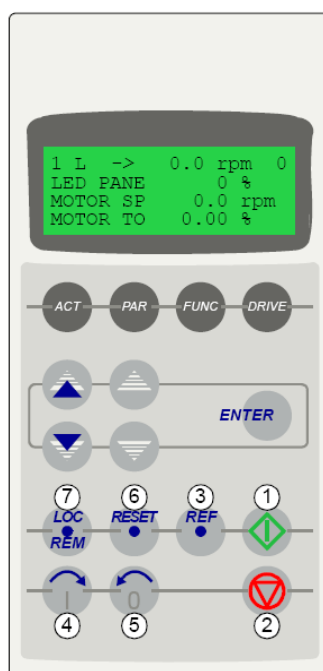


### Ovládání měniče pomocí panelu

Panel CDP 312 je vybaven čtyřřádkovým displejem a několika ovládacími tlačítky (Obr. 6). Je možné si vybrat ze dvanácti světových jazyků, mezi nimiž je i čeština (výběr jazyka se provádí pomocí tlačítka FUNC). Ovládání panelu je poměrně snadné a intuitivní. Pro aktivaci ovládání měniče z panelu musíme zmáčknout tlačítko č.7. Poté se v levém horním rohu displeje objeví písmeno L (LOC), což znamená, že jsme přepnuli do lokálního ovládání. Pokud totiž v tomto rohu není zobrazeno žádné písmeno, je měnič ovládán vzdáleně (REM). Tlačítkem RESET (č.6) vymažeme případné chyby vzniklé chodem měniče. Použitím tlačítka REF (č.3) můžeme zadat referenci otáček. Při jeho stisknutí aktivujeme políčko pro zadání reference (v pravém rohu displeje). Nastavení hodnoty provedeme pomocí šipek na panelu a následně hodnotu potvrdíme pomocí tlačítka ENTER. Tlačítka č.4 a č.5 umožňují nastavit směr otáčení hřídele motoru. Zeleným tlačítkem (č.1) motor spustíme a červeným (č.2) zastavíme.

Ovládací panel má čtyři provozní režimy:

- Režim zobrazení aktuálního signálu (tlačítko ACT)
- Parametrický režim (tlačítko PAR)
- Režim funkcí (tlačítko FUNC)
- Režim výběru pohonu (tlačítko DRIVE)



Obr. 6 Panel CDP 312

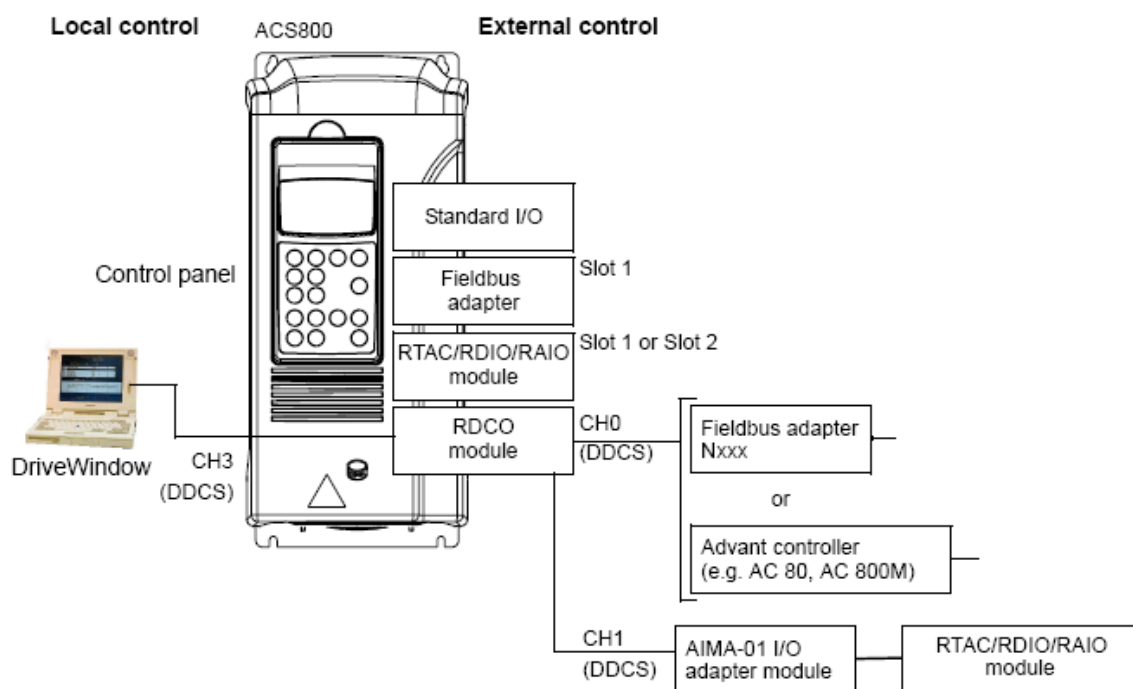
K dispozici jsou tyto aktuální signály:

- výstupní frekvence, proud, napětí a výkon měniče,
- otáčky a točivý moment motoru,
- síťové napětí a napětí ve stejnosměrném meziobvodu,
- aktivní řídicí místo (místní, EXT1 nebo EXT2),
- referenční hodnoty,
- teplota měniče,
- počítadlo provozních hodin (hod), počítadlo kWh,
- stav digitálních a analogových vstupů a výstupů,
- PID regulátor aktuální hodnoty, (pokud je zvoleno makro PID regulace).

Na displeji ovládacího panelu mohou být současně zobrazeny hodnoty tří zvolených signálů.

**Externí řízení**

Měnič může být ovládán externími zařízeními přes komunikační síť. Připojení se provádí pomocí adaptérových modulů upevněných v rozšiřujícím slotu měniče. Následující obrázek znázorňuje způsob zapojení a rozdíl mezi lokálním a externím řízením.



**Obr. 7 Lokální a externí ovládání [ABB 2007]**



V mém případě měnič obsahuje tyto moduly:

- PROFIBUS DP Adapter Module RPBA-01
- Pulse Encoder Interface Modul RTAC-01
- Komunikační modul RDCO

Modul PROFIBUS DP RPBA-01 slouží pro připojení a komunikaci přes rozhraní ProfiBus. Je v měniči umístěn úplně nahoře a je osazen dvěma LED diodami pro signalizaci komunikace.

Pulse Encoder Interface Modul slouží jako rozhraní pro digitální pulzní kodér. Ten může být použit v případě, kdy vyžadujeme zpětnou vazbu na přesnou rychlost nebo polohu (úhel natočení) hřídele motoru. Ve výchozím nastavení se totiž otáčky počítají z matematického modelu, kterým měnič disponuje. Ale i ten je dostatečně přesný. Rozdíly jsou patrné jen v nižších otáčkách a nejsou nikterak výrazné.

Komunikační modul RDCO (optické rozhraní) disponuje celkem čtyřmi kanály pro připojení (CH0, CH1, CH2, CH3). Přičemž každý kanál je určen pro jiná zařízení. Rozdělení kanálů podle účelu je následující:

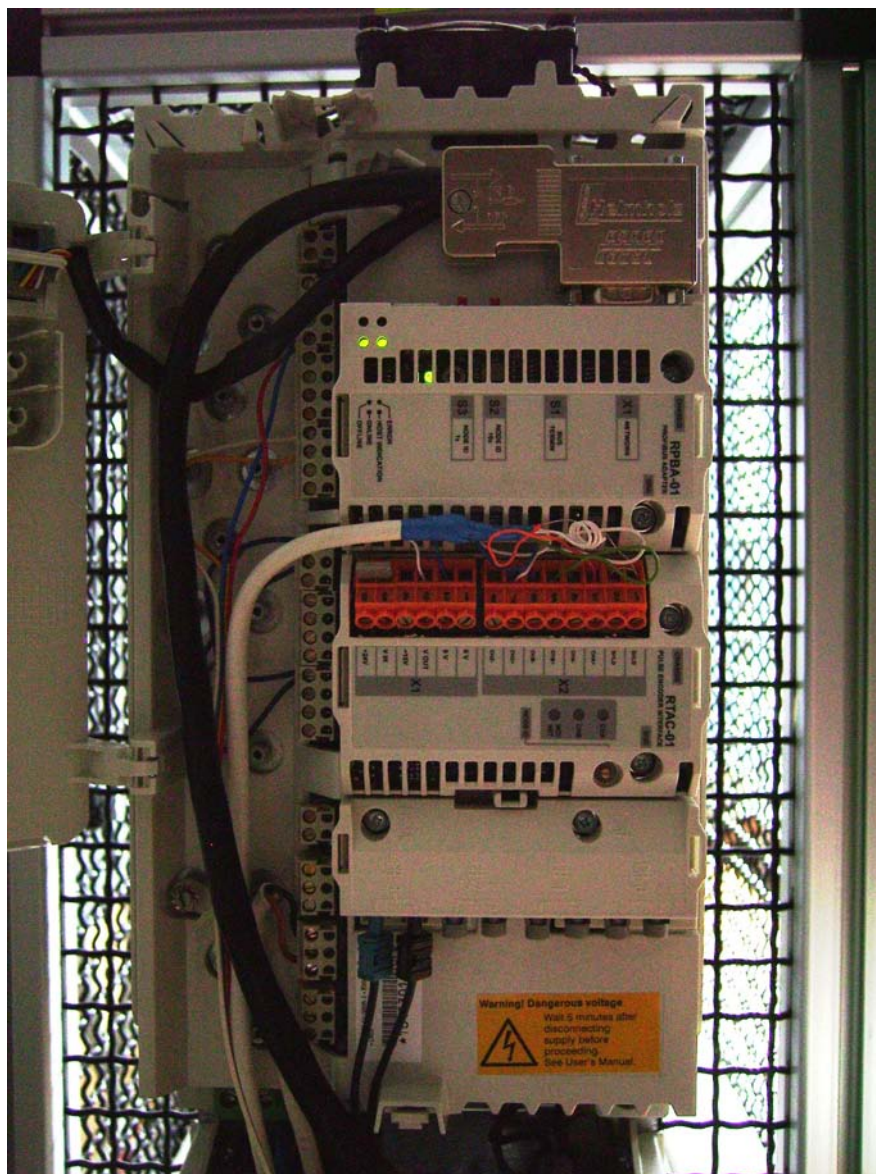
- CH0 – nadřazený řídicí systém (PLC), nebo další moduly
- CH1 – jednoduché a volitelné I/O
- CH2 – Master / Follower
- CH3 – DriveWindow (PC)

Podrobnější popisy modulů, zejména pak modulů RPBA-01 a RTAC-01 najdete v příslušných manuálech. Ty jsou umístěny i na CD dodaném k zařízení. Nebo k dispozici na internetu.



rozhraní spojuje PC přímo s frekvenčním měničem přes optický ModuleBus. Toto propojení slouží k ovládání měniče použitím softwarového nástroje DriveWindow. Počítač slouží jak pro práci s aplikacemi obsluhujícími PLC a frekvenční měnič, tak pro následnou vizualizaci programem InTouch.

Na fotografii níže můžete vidět umístění adaptérových modulů připojených do rozšiřujícího slotu měniče.

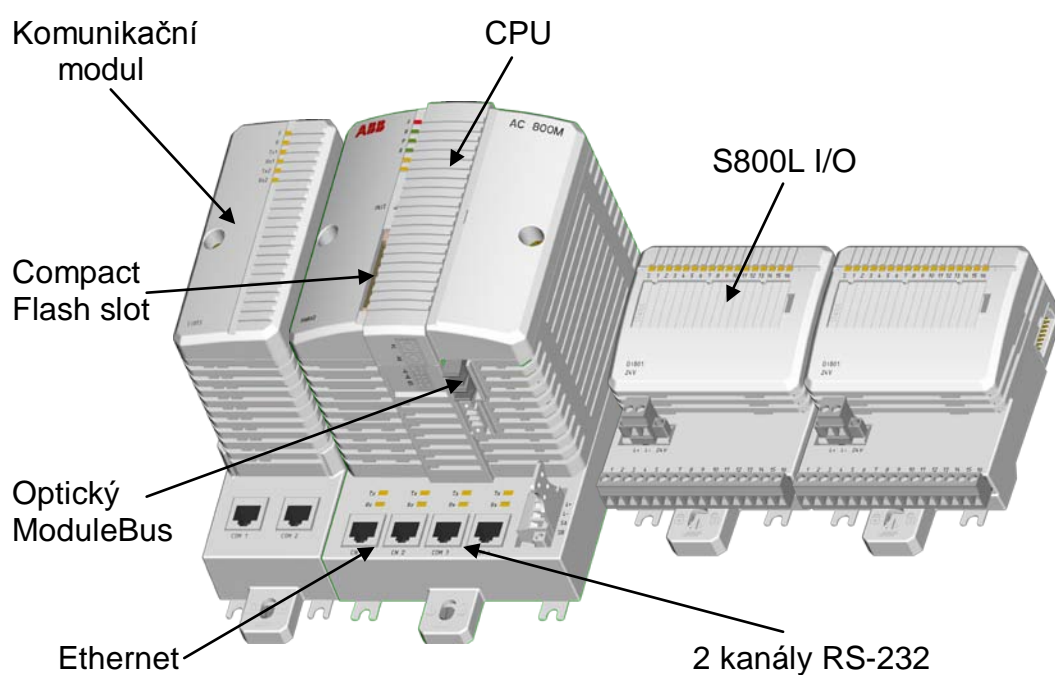


Obr. 9 Umístění adaptérových modulů

## 4.1 PLC

K řízení měniče je možné použít také PLC. Na laboratoři F204 je umístěn systém AC 800M. Jedná se o modulární, škálovatelný řídicí systém od firmy ABB. Hodí se jak pro jednoduché aplikace, tak i pro rozsáhlé projekty. Umožňuje aplikovat jak základní logické funkce, tak i komplexní typy regulátorů včetně regulace pomocí FUZZY logiky.

Vývojovým prostředím umožňujícím programování a konfiguraci tohoto systému je Compact Control Builder (někdy označováno také jako PLC Control Builder AC 800M). Popisu tohoto programu se věnuje kapitola č.6.

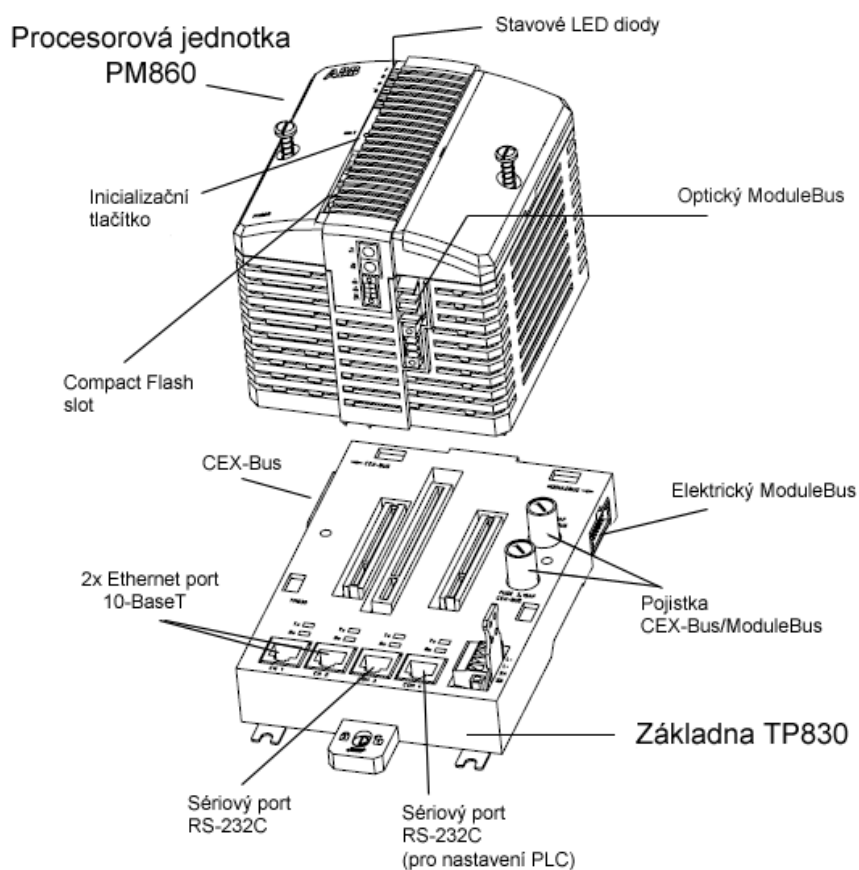


Obr. 10 AC 800M [ABB – upraveno]

Systém AC 800M se skládá z procesorové jednotky a komunikačních modulů. Vše je určeno k montáži na lištu DIN. K procesorové jednotce je možné připojit až 12 komunikačních modulů přes sběrnici CEX. Dále je možné připojit různé I/O systémy (moduly S800 – maximálně 12) přes elektrický ModuleBus. Třetí možností připojení je optický ModuleBus. Tato sběrnice umožňuje připojení až 7 I/O skupin, přičemž každá skupina může obsahovat až 12 modulů S800. To znamená, že přes optický ModuleBus je možné připojit celkem až 84 I/O modulů. V našem případě si však vystačíme se sběrnicemi procesorové jednotky.

Procesorová jednotka systému AC 800M je přímo osazena celkem třemi komunikačními rozhraními. Dvakrát rozhraní ethernet, dvakrát RS232 a optický ModuleBus. Pro naši úlohu použijeme rozhraní ethernet pro připojení PC. Pro připojení frekvenčního měniče použijeme komunikační modul CI854A (ProfiBus).

Firma ABB vyrábí pro systém AC 800M několik typů procesorových jednotek. Liší se podle frekvence procesoru (24MHz – 96MHz) a velikosti paměti (8MB – 32MB). V našem případě jde o procesorovou jednotku PM860, jejíž procesor má frekvenci 48MHz a paměť velikost 8MB. Jednotka je připevněna k základně TP830, která obsahuje porty komunikačních rozhraní (Obr. 11).

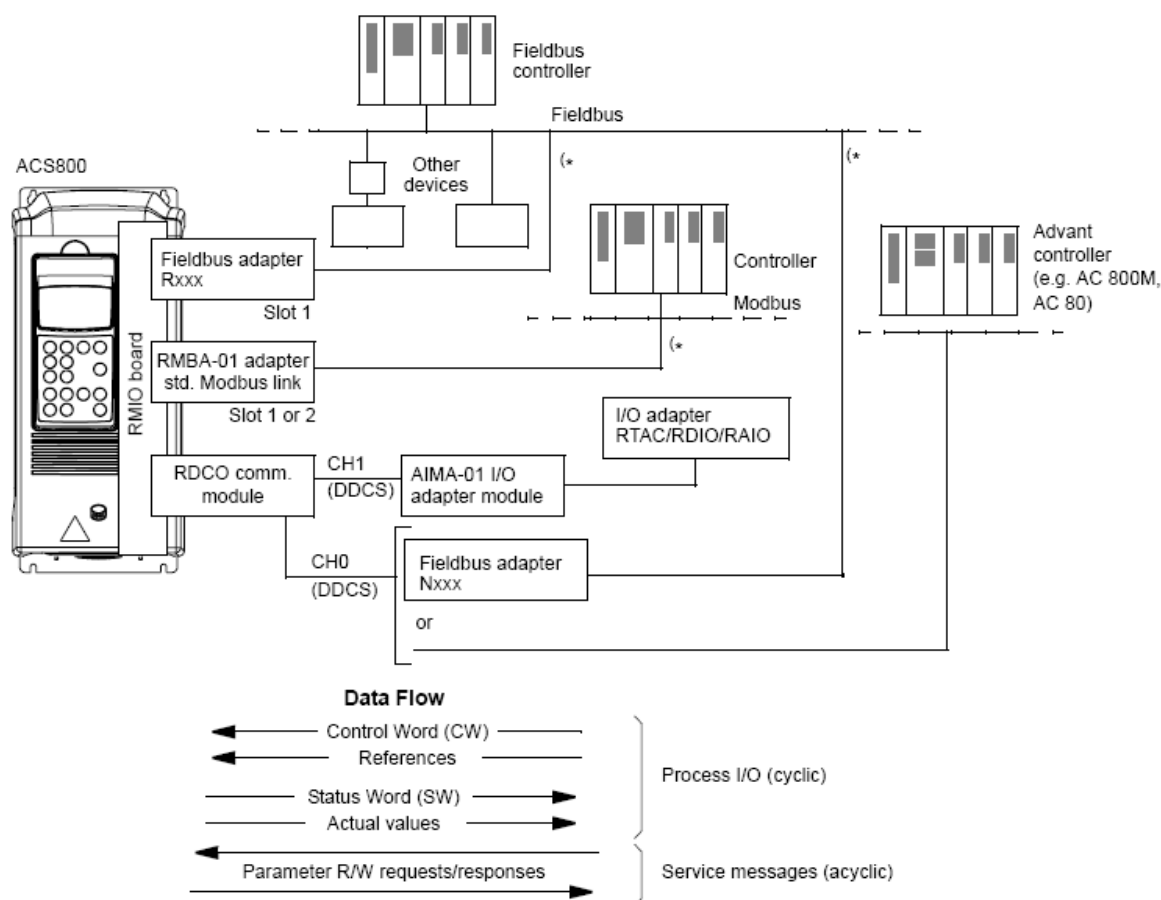


Obr. 11 Procesorová jednotka PM860 a základna TP830 [ABB]

## 4.2 Nastavení měniče

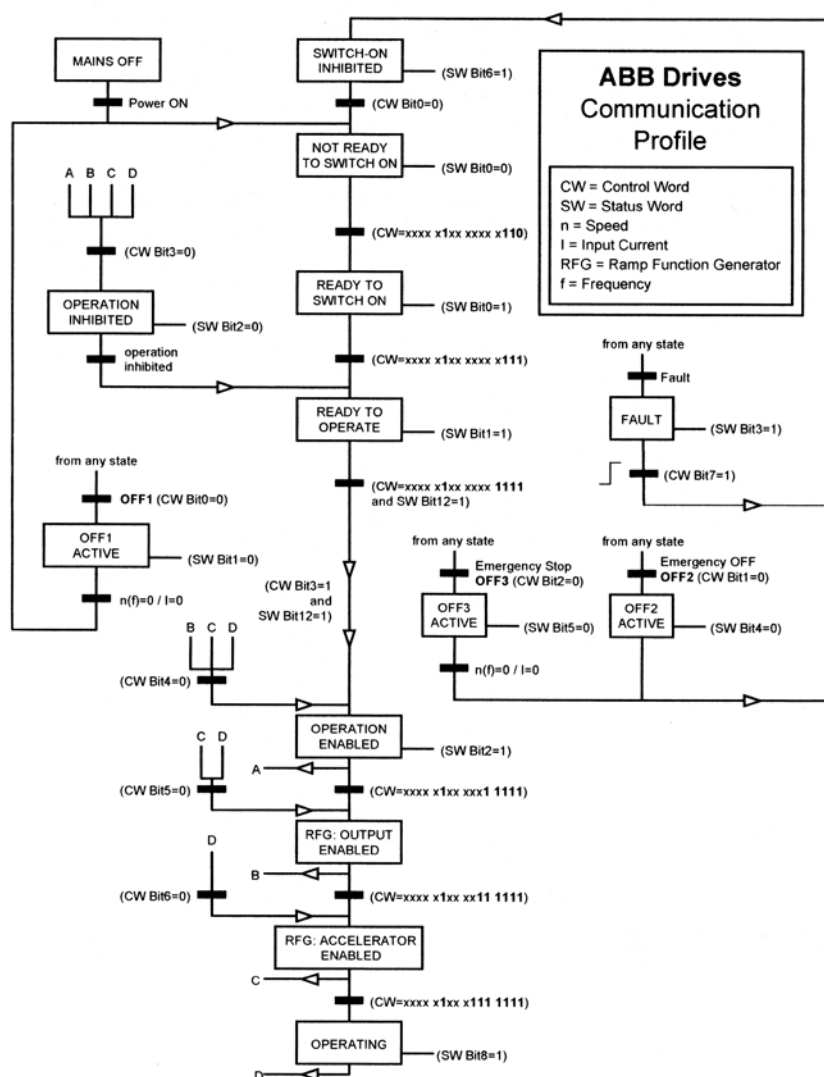
I když je úloha propojena nezávisle pomocí dvou rozhraní, měnič je schopen v jednom čase přijímat řídicí příkazy jen přes jedno z nich. Je to způsobeno tím, že ovládání měniče pomocí PC (DriveWindow) je bráno jako zvláštní způsob lokálního ovládání. Což je na první pohled trochu matoucí, ale vše vysvětluje obrázek Obr. 7. Tudíž je nutné tuto komunikaci nastavit v parametrech měniče. Jak už jsem zmínil, použil jsem jako výchozí rozhraní profibus, k čemuž je na měniči umístěn PROFIBUS DP Adapter Module RPBA-01.

Tento modul je před vlastním použitím nutné aktivovat nastavením parametrické sady 98.02:COMM. MODULE – zvolit FIELDBUS a v parametrické sadě 98.08:COMM PROFILE – nastavit ABB DRIVES. Poté se aktivuje parametrická sada 51, konkrétně 51.01:MODULE TYPE – PROFIBUS DP, tím poznáme, že je modul aktivní a použit pro řízení. Dále je nutné nastavit par. 51.02:Node address – adresa uzlu nastavená přepínačem na modulu. Podrobnější nastavení měniče viz Firmware Manual.



Obr. 12 [ABB 2007]

Předcházející schéma znázorňuje možné způsoby připojení měniče k nadřazenému řídicímu systému (PLC) a princip komunikace – tok dat (Data Flow). Princip této komunikace je následující: Měnič je řízen signály („slovy“) Control Word = řídicí slovo. Pomocí těchto slov zadáváme měniči příkazy (reference). Naproti tomu měnič dává vědět o svém stavu vysíláním signálu Status Word = statusové slovo. Prostřednictvím výměny těchto dvou slov probíhá komunikace. Slova se skládají z šestnácti bitů, z nichž každý má svůj význam. Popis významu jednotlivých bitů uvádí tabulky na dalších dvou stranách.



Obr. 13 Komunikační profil ABB pohony [ABB 2004]

Schéma výše znázorňuje komunikační profil ABB pohony. Představuje postup, kterým lze motor spustit a následně ovládat. Důležité jsou jednotlivé podmínky, které udávají, jaké bity musí být sepnuty (nastaveny na hodnotu 1). Přičemž záleží na pořadí spínání jednotlivých bitů, viz schéma. Při nedodržení tohoto pořadí nebude motor reagovat.

Tab. 1 Control Word

Bit	Název	Hodn.	Zadejte stav / popis
0	OVLADANI VYP1  (OFF1 CONTROL)	1	Zadejte <b>REDY TO OPERATE</b> (připraven k činnosti).
		0	Zastavení po současné aktivní decelerační rampě. Zadejte <b>OFF1 ACTIVE</b> (aktivní VYP1); postupujte na <b>READY TO SWITCH ON</b> (připraven k zapnutí), pokud nejsou vzájemná blokování (VYP2, VYP3) aktivní.
1	OVLADANI VYP2  (OFF2 CONTROL)	1	Pokračujte v provozu (VYP2 neaktivní).
		0	Nouzové VYP. Dobíhá, až se zastaví; Zadejte <b>OFF2 ACTIVE</b> (VYP2 aktivní) postupujte na <b>NOT READY TO SWITCH ON</b> (zabráněno zapnutí).
2	OVLADANI VYP3  (OFF3 CONTROL)	1	Pokračujte v provozu (VYP3 neaktivní).
		0	Nouzové zastavení, zastavení během doby definované parametrem. Zadejte <b>OFF3 ACTIVE</b> (VYP3 aktivní); postupujte na <b>NOT READY TO SWITCH ON</b> (zabráněno zapnutí). <b>Varování:</b> Zajistěte, aby motor a hnací stroj mohly být zastaveny s použitím tohoto režimu zastavení.
3	ZABRANENO ČINNOSTI (INHIBIT_OPERATION)	1	Zadejte <b>OPERATION INHIBITED</b> (zabráněno činnosti).
		0	0 Blokuje provoz. Zadejte <b>OPERATION INHIBITED</b> (zabráněno činnosti).
4	RAMP_VYST_NULA  (RAMP_OUT_ZERO)	1	Normální provoz. Zadejte <b>RGF: OUTPUT ENABLED</b> (RFG: výstup uvolněn).
		0	Přinutí výstup generátoru rampové funkce přejít na nulu. Pohon se zastaví po rampě. Platí limity proudu DC napětí).
5	RAMP_PRIDRZ  (RAMP_HOLD)	1	Povoluje funkci rampy. Zadejte <b>RGF: ACCELERATOR ENABLED</b> (RFG: urychlovač uvolněn).
		0	Zastaví činnosti rampy (zachová výstup generátoru rampové funkce).
6	RAMP_DO_NULY (RAMP_IN_ZERO)	1	Normální provoz. Zadejte <b>OPERATING</b> (v činnosti).
		0	Přinutí vstup generátoru rampové funkce přejít na nulu.
7	RESET	0 → 1	Porucha resetování, jestliže existuje aktivní porucha. Zadejte <b>SWITCH ON INHIBITED</b> (zabráněno zapnutí).
		0	Pokračujte v normálním provozu.
8	INCHING_1	1	Nepoužívá se.
		1 → 0	Nepoužívá se.
9	INCHING_1	1	Nepoužívá se.
		1 → 0	Nepoužívá se.
10	DALK_RIZENI  (REMOTE_CMD)	1	Ovládání fieldbusu povoleno.
		0	Řídicí slovo <> 0 nebo Reference <> 0: Zachová poslední řídicí slovo a reference. Řídicí slovo = 0 a Reference = 0: Ovládání fieldbusu povoleno. Reference a decelerační/akcelerační rampy jsou zamčeny.
11	EXT MISTO RIZ (EXT CTRL LOC)	1	Vybrat Externí řídicí místo EXT2. Účinné, jestliže parametr je nastaven na KOMUN.RS(11).
		0	Vybrat Externí řídicí místo EXT1. Účinné, jestliže parametr je nastaven na KOMUN.RS(11).
12-15	Rezervováno		

Dle typu požadovaného programu je možné měniči posílat buď jednotlivé bity, nebo můžeme celé slovo (16 bitů) převést z binární podoby do decimálního tvaru a poslat toto decimální číslo. Například pokud posloupnost bitů 1000111111 převedeme do decimální podoby, dostaneme číslo 1151. Je tedy možné měniči poslat toto číslo (1151). Po odeslání tohoto slova (nebo čísla) dojde k rozběhu pohonu (viz schéma na předchozí straně).

Následující tabulka obsahuje významy jednotlivých bitů slova „Status Word“, které měnič posílá do nadřazeného systému.



Tab. 2 Status Word

Bit	Název	Hodn.	Stav / popis
0	RDY_ON	1	<b>READY TO SWITCH ON</b> (připraven k zapnutí)
		0	<b>NOT READY TO SWITCH ON</b> (nepřipraven k zapnutí)
1	RDY_RUN	1	<b>REDY TO OPERATE</b> (připraven k činnosti)
		0	<b>OFF1 ACTIVE</b> (VYP1 aktivní)
2	RDY_REF	1	<b>OPERATION ENABLED</b> (činnost povolena)
		0	<b>OPERATION INHIBITED</b> (zabráněno činnosti)
3	TRIPPED	1	<b>FAULT</b> (porucha)
		0	Žádná porucha
4	OFF_2_STA	1	VYP2 neaktivní
		0	<b>OFF2 ACTIVE</b> (VYP2 aktivní)
5	OFF_3_STA	1	VYP3 neaktivní
		0	<b>OFF3 ACTIVE</b> (VYP3 aktivní)
6	SWC_ON_INHIB	1	<b>SEITCH ON INHIBITED</b> (zabráněno zapnutí)
		0	
7	ALARM	1	Varování/alarm
		0	Žádné varování/alarm
8	AT_SETPOINT	1	<b>OPERATING</b> (v činnosti). Aktuální hodnota se rovná referenční hodnotě (= je v rámci limitů tolerance)
		0	Aktuální hodnota se liší od referenční hodnoty (= je mimo limity tolerance)
9	REMOTE	1	Řídicí místo pohonu: REMOTE (EXT1 nebo EXT2). (DÁLKOVÉ)
		0	Řídicí místo pohonu: LOCAL (MÍSTNÍ)
10	ABOVE_LIMIT	1	Externí řídicí místo EXT2 vybráno
		0	Externí řídicí místo EXT1 vybráno
11	EXT RUN ENABLE	1	Externí signál (CHOD POVOLEN) obdržen
		0	Žádný externí signál (CHOD POVOLEN) nebyl obdržen
12-14	Rezervováno		
15		1	Komunikační chyba detekovaná fieldbusovým adaptérovým modulem (na optickém kanálu CH0)
		0	Komunikace fieldbusového adaptéru (CH0) je v pořádku

## 5 Použitý software

Pro danou úlohu je zapotřebí celá řada programů sloužících k ovládání, programování či vizualizaci dané technologické jednotky. Všechny použité programy vyžadují operační systém Windows. Minimálním požadavkem pro instalaci je operační systém Windows 2000 nebo Windows XP Professional. Dále je dobré mít nainstalován Microsoft Word 2000 nebo novější a také Acrobat Reader verze 4.0 nebo vyšší. Microsoft Word je potřebný pro vytváření uživatelsky definovaných projektových dokumentací a Acrobat Reader ke čtení online manuálů. Aplikační server Wonderware Application Server, který je součástí platformy Wonderware System Platform 3.0, vyžaduje navíc Microsoft SQL Server 2005 Service Pack 2. Přičemž je důležité použít přesně tuto verzi. Alespoň v době psaní tohoto textu to bylo nutné. Zkoušel jsem totiž použít SQL Server 2008, ale ten s Application Server nepracoval. Při pokusu o spuštění IDE, docházelo k zamrznutí celé aplikace. Problém vyřešil až návrat k SQL Serveru 2005 SP2.

Mezi použité programy patří:

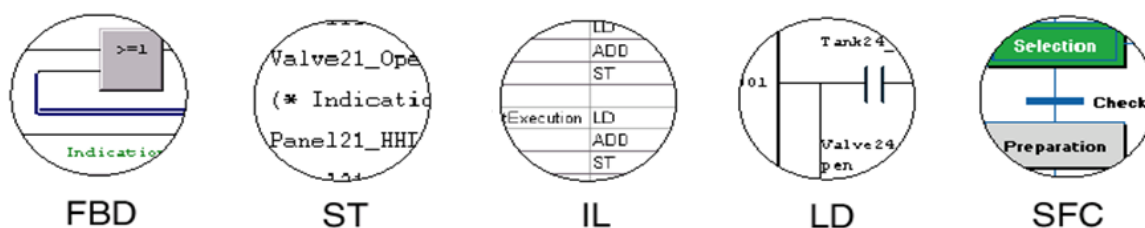
- PLC Control Builder AC 800M
- OPC Server for AC 800M (verze 4.1)
- SoftController 4.1
- DriveWindow 2.12
- InTouch 10.0
- Wonderware Application Server 3.0
- OPCLink 8.0
- FS Gateway

Dále podrobněji představím každý z nich, přičemž také uvedu hardwarové požadavky pro instalaci a provoz aplikací. Případně zmíním zkušenosti s instalací. V mém případě jsem instalaci prováděl pod operačním systémem MS Windows XP Professional Service Pack 3.

## 5.1 PLC Control Builder AC 800M

PLC Control Builder je programovací nástroj společnosti ABB, pro vytváření řídicích řešení založených na PLC. Existují různé verze tohoto softwaru, které se liší podle toho, pro jaký hardware (PLC) je software určen (v mém případě je použito PLC AC 800M). V souladu s mezinárodní normou IEC 61131-3 podporuje Control Builder celkem pět různých programovacích jazyků. Volba programovacího jazyka se odvíjí od zkušeností programátora a účelu programu. Následující řádky uvádí přehled programovacích jazyků a jejich stručnou charakteristiku (jazyky jsou uvedeny v pořadí, jak byly vyvíjeny):

1. **LD (Ladder Diagram)** – *kontaktní schéma*; výhodou bylo, že tuto techniku ovládali všichni elektroprojektanti navrhující reléové ovládací obvody; u této jednoduché formy zápisu musí automat soustavně vyhodnocovat všechny příkazy, i ty, které v dané chvíli vyhodnocovány být nemusí.
2. **FBD (Function Block Diagram)** – *funkční blokové schéma*; stejný původ jako metoda LD, jen předpokládá projektanty znalé projektování ovládacích schémat z integrovaných prvků realizujících základní logické funkce.
3. **IL (Instruction List)** – *seznam příkazů*; byl užíván u nejprimitivnějších automatů; jedná se v podstatě o jazyk symbolických adres, tzv. assembler.
4. **ST (Structured Text)** – *strukturovaný text*; vyšší programovací jazyk, podobný PASCALu
5. **SFC (Sequential Function Chart)** – *sekvenční funkční diagram*; vychází z GRAFCETu, je prakticky totožný s jeho průmyslovou variantou; metoda sama o sobě může být použita jen k hrubší analýze nebo k hrubému návrhu algoritmů, plnohodnotná metoda vznikne, až když je možné zapsat algoritmickou náplň jednotlivých kroků v nějakém programovacím jazyce (ST); v současnosti je kombinace SFC a ST považována za ideální metodu programování PA.



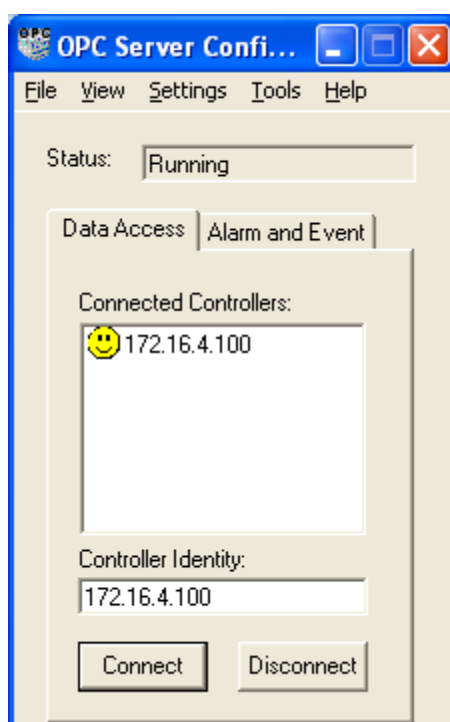
Obr. 14 Náhled programovacích jazyků

Jádrem rozhraní Control Builderu je Project Explorer, ten zobrazuje aktivní projekt. Prostředí Control Builderu podrobně popisuje další kapitola (č.6). Komunikace mezi Control Builderem, OPC serverem a PLC probíhá pomocí standardu MMS (Manufacturing Message Specification). MMS je mezinárodní protokol pro výměnu real-time dat mezi propojenými zařízeními nebo počítačovými aplikacemi.

## 5.2 OPC Server for AC 800M 4.1

OPC je otevřený standard pro průmyslovou automatizaci, který specifikuje komunikaci mezi softwarovými aplikacemi nebo zařízeními. Je založen na architektuře klient/server.

OPC server slouží pro výměnu dat z různých zdrojů (PLC, databáze, softwarové aplikace aj.) používající různé typy připojení (sériové, ethernet, rádiový přenos) a různý operační systém (Windows, DOS, UNIX, VMS). Před instalací OPC Serveru je nutné mít nainstalován PLC Control Builder. OPC server při instalaci totiž potřebuje načíst nastavení Control Builderu.



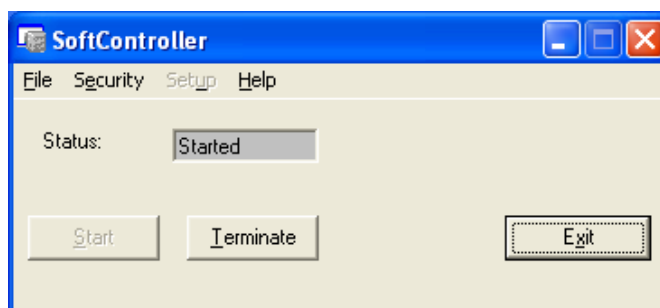
Obr. 15 OPC Server

Aby bylo možné propojit PLC s OPC Serverem musíme nastavit IP adresu dostupného PLC. Tu napíšeme do pole Control Identity a klikneme na tlačítko Connect. Poté se v okně Connected Controllers objeví status připojení (žlutý smajlík a IP adresa PLC). Pokud bychom chtěli místo reálného PLC použít SoftController, musíme nejprve spustit simulaci

v okně aplikace SoftController. V opačném případě se připojení nepodaří a status bude indikován jako nedostupný (červené kolečko s bílým křížkem). V případě, kdy používáme SoftController, běžící na stejném počítači jako Control Builder je IP adresa pro PLC stejná jako IP adresa počítače, s tím rozdílem, že na konec adresy musíme přidat dvojtečku a číslici 2 (např. 172.16.4.100:2).

### 5.3 SoftController 4.1

Pomocí tohoto programu můžeme aplikaci vytvořenou v PLC Control Builderu odsimulovat bez připojení k reálnému PLC. To je výhodné, pokud chceme navrženou aplikaci nejprve otestovat, a nebo v situaci, kdy nemáme k PLC přístup. SoftController spustíme kliknutím na tlačítko start. Běh programu je indikován nápisem „Started“ v poli Status.



Obr. 16 SoftController

### 5.4 DriveWindow 2.12

DriveWindow je jednoduchá aplikace sloužící ke spuštění a údržbě řídicích jednotek (frekvenčních měničů) ABB s komunikací přes optický kabel. DriveWindow je nástroj založený na standardu OPC. Obsahuje DriveOPC (OPC server pro DDCS síť). Pomocí DriveWindow můžeme měnit parametry pohonu (nahrát parametry do měniče) a porovnávat nastavení s různými parametry. Umožňuje také sledovat časovou závislost parametrů a měřených veličin v reálném čase (monitoring). Pro instalaci tohoto softwaru je potřeba operačního systému Windows NT, 2000 nebo XP. Přehled hardwarových požadavků a funkcí DriveWindow uvádím níže.

Po spuštění DriveWindow je nutné v dialogovém okně zvolit OPC Server. V našem případě vybereme server ABB.SMP. Pokud je hardware připojen k měniči správně, začne DriveWindow s měničem komunikovat. V případě neúspěšného připojení se zobrazí

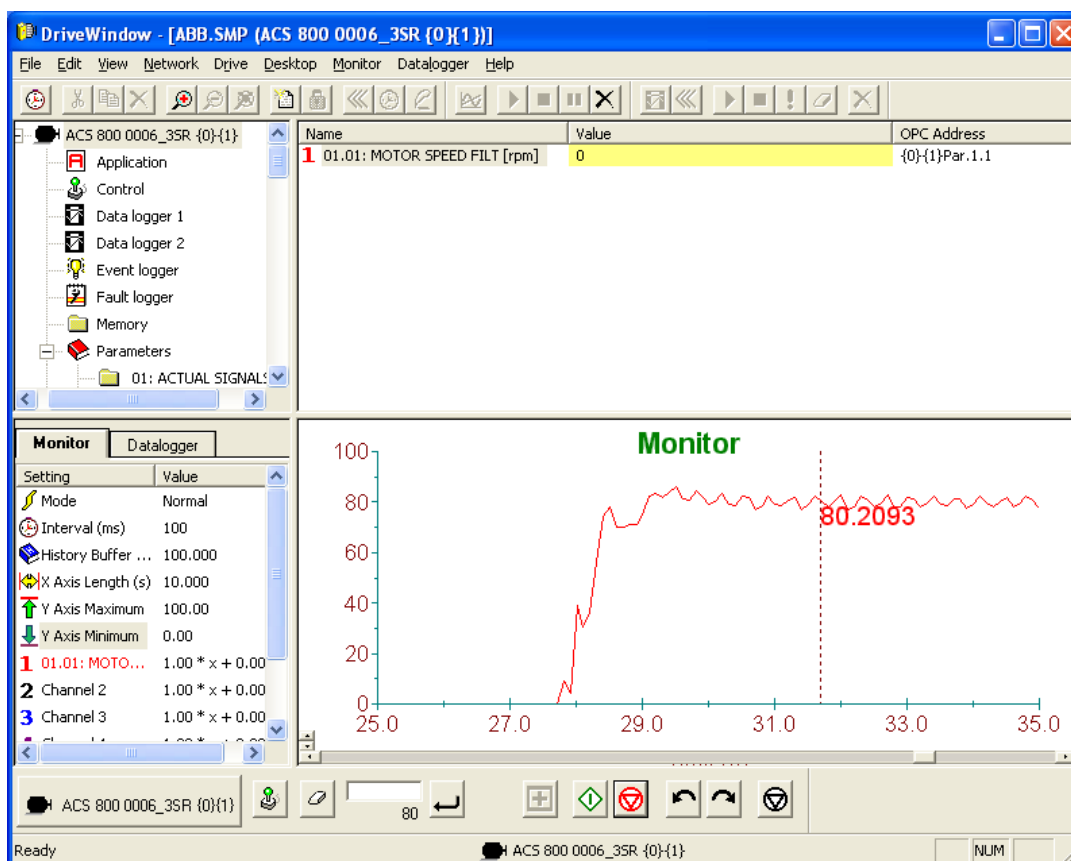
chybová zpráva a je nutné připojený hardware zkontrolovat. Měnič musí být nastaven na ovládání pomocí vzdálené komunikace (viz. *Ovládání měniče pomocí panelu*, str. 14).

#### Hardwarové požadavky:

- PCMCIA karta NDPC-12 (PCMCIA slot nebo PCI slot s adaptérem PCI/PCMCIA)
- Síťová karta (pokud chceme použít vzdálené připojení)
- DDCS kabel (optický kabel)

#### Funkce DriveWindow:

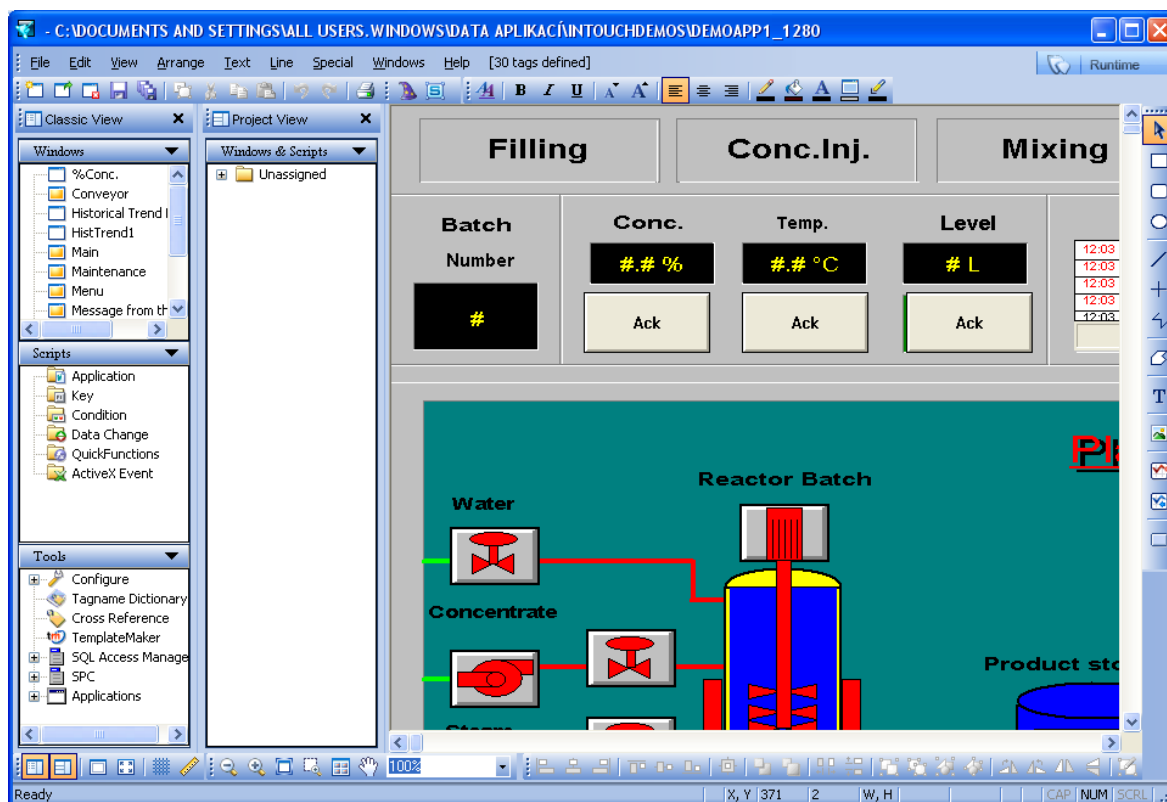
- Řídící operace (spuštění, zastavení, reference atd.)
- Monitorování (signálů)
- Parametry (změna hodnot)
- Zápis chyb (zobrazení chyb)
- Záloha a obnova (kompletní záloha softwaru)
- Všechny funkce dostupné přes síť (řízení, monitorování atd.)



Obr. 17 Uživatelské prostředí DriveWindow

## 5.5 InTouch 10.0

Jedná se o softwarový systém kategorie SCADA/HMI (Supervisory Control and Data Acquisition / Human Machine Interface) pro vizualizaci, sběr dat a supervizní řízení technologických a výrobních procesů. Program InTouch umožňuje vytvořit grafické zobrazení jakýchkoli výrobních technologií na monitoru počítače, jejich ovládání a dynamické animace, které v reálném čase zobrazují aktuální stavy provozních systémů. Verze 10.0 představuje nejvýznamnější funkční vylepšení produktu od doby, kdy byl poprvé uveden na trh. Nová verze přináší průlomovou architekturu a nové grafické možnosti technologie ArchestrA. Toto nové řešení kombinuje systémy Wonderware System Platform 3.0 a InTouch 10.0. Obsahuje tradiční vývojové prostředí WindowMaker a navíc integrované vývojové prostředí ArchestrA IDE (Integrated Development Environment) společné pro vývoj aplikací v InTouch 10.0 i pro vývoj řešení na bázi platformy Wonderware System Platform 3.0. Tato integrace umožňuje snadný přechod pro uživatele tradičních aplikací InTouch k aplikacím na bázi serverové platformy. Využití platformy Wonderware System Platform 3.0 není pro provozování aplikací InTouch nutné, avšak přináší zásadní rozšíření možností softwaru InTouch. Mezi tato rozšíření patří například ArchestrA grafika a aplikační objekty, centralizovaná správa aplikací aj.



Obr. 18 Prostředí programu InTouch 10.0

Díky integraci InTouch s ArchestrA mohou být aplikace spravovány pomocí rozhraní InTouch Application Manager nebo vývojového prostředí ArchestrA IDE. Tímto se aplikace InTouch dělí na tři typy, podle toho, jakým způsobem jsou spravovány nebo jak jsou publikovány.

#### Kategorie aplikací InTouch:

- **Klasické aplikace** – jsou spravovány pomocí programu InTouch Application Manager, lze je provozovat bez využití systému Wonderware System Platform ale zároveň není možné využít nové ArchestrA grafiky, v programu InTouch Application se objevují jako „Stand Alone“ (klasické)
- **Aplikace spravované z IDE** – spravované použitím vývojového prostředí ArchestrA IDE, založené na Wonderware System Platform, aplikace InTouch jsou využity jako klientské aplikace pro Wonderware System Platform, je nutné použít serverové komponenty (Wonderware Application Server), tyto aplikace jsou nazývány „Managed“ (spravované), aplikace se vyvíjí v jednom uzlu databáze Galaxy společně s programem WindowMaker, mohou využít všech možností nové architektury včetně ArchestrA grafiky
- **Aplikace publikované z IDE** – aplikace spravované z IDE lze po vytvoření publikovat, tak aby mohly být distribuovány jako klasické InTouch aplikace

#### Konektivita

InTouch lze připojit prakticky k jakýmkoliv průmyslovým automatizačním řídicím nebo informačním zařízením (PLC, I/O systémy atd.) prostřednictvím různých komunikačních programů, tzv. I/O nebo OPC serverů, které jsou k dispozici od firmy Wonderware, výrobců řídicích systémů nebo dalších dodavatelů. InTouch může fungovat jako OPC klient i OPC server.

Komunikační servery od Wonderware podporují následující komunikační standardy:

- Technologie OPC
- Protokol Wonderware SuiteLink
- Protokol Microsoft DDE (Dynamic Data Exchange)

K dispozici je navíc sada vývojářských nástrojů ArchestrA DAS Toolkit, která umožňuje vývoj vlastních komunikačních serverů pro nová, atypická nebo doposud nepodporovaná zařízení.

[Wonderware]



## 5.6 Wonderware Application Server 3.0

Aplikační server pro průmyslové automatizační a informační aplikace. Dříve byl dodáván samostatně pod názvem Industrial Application Server (IAS). Současná verze aplikačního serveru je dodávána jako součást serverového softwarového systému Wonderware System Platform 3.0.

Wonderware System Platform 3.0 je softwarová platforma pro průmyslové aplikace, založená na technologické infrastruktuře ArchestrA. Jedná se o ucelenou softwarovou architekturu pro efektivní navrhování, provoz a údržbu průmyslových automatizačních a informačních systémů určených pro:

- Vizualizační a supervizní aplikace kategorie HMI
- Supervizní a vizualizační aplikace typu SCADA/GeoSCADA, včetně rozsáhlých geograficky rozptřených technologických celků
- Aplikace pro vyspělé řízení a analýzu výrobních operací kategorie Production and Performance Management (řízení výroby a analýza výkonnosti výroby), resp. MES (Manufacturing Execution Systems)

Systémy Wonderware System Platform 3.0 jsou dodávány jako integrované softwarové sady, které obsahují následující aktuální verze serverových produktů od firmy Wonderware:

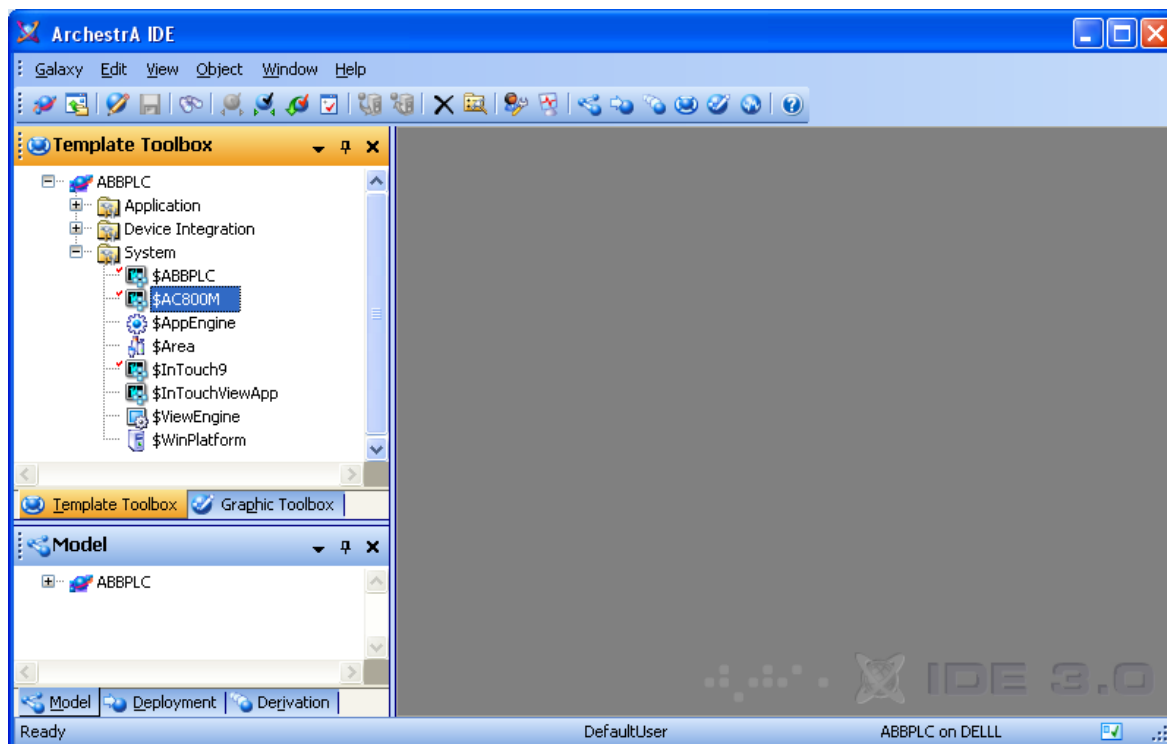
- Wonderware Application Server 3.0 – průmyslový aplikační server
- Wonderware Historian 9.0 – procesní historizační databáze (předchozí název IndustrialSQL Server)
- Wonderware Information Server 3.0 – výrobní informační portál (předchozí název SuiteVoyager)
- Wonderware komunikační Servery – I/O nebo DA Servery pro komunikaci s řídicími zařízeními (PLC aj.)

Wonderware Application Server zajišťuje v reálném čase robustní komunikaci s řídicími systémy, sběr dat a generování alarmů, provádí archivaci dat do historizační databáze a umožňuje centrální nasazení aplikací a správu zabezpečení.

[Pantek (CS) s.r.o.]

### **Integrované vývojové prostředí ArchestrA IDE**

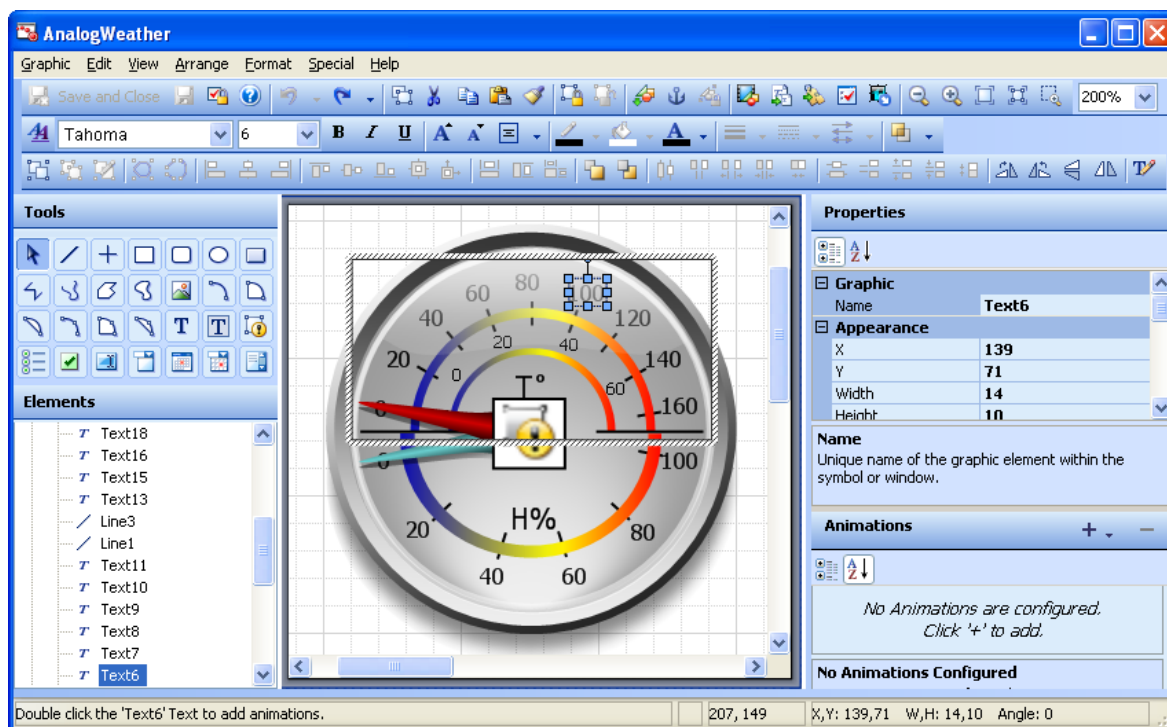
Modernizované vývojové prostředí ArchestrA IDE bylo přepsáno s využitím platformy Microsoft .NET Framework 2.0. Tato změna přinesla nové vlastnosti jako plovoucí panely nástrojů s možností automatického zobrazení/skrytí, které vývojáři umožňují efektivní využití pracovní plochy v IDE. Vývojové prostředí ArchestrA IDE podporuje spolupráci více vývojářů na jednom projektu. K vizualizaci řízených procesů využívá Wonderware Application Server vizualizační software InTouch.



**Obr. 19** Integrované vývojové prostředí ArchestrA IDE

Jednou z hlavních novinek InTouch 10.0 je objektově orientovaná ArchestrA grafika, která je dostupná při použití integrovaného vývojového prostředí ArchestrA IDE. Nová grafika je reprezentována novým typem symbolů pojmenovaných jako ArchestrA Symbols. Což jsou inteligentní grafické symboly, ve kterých každý základní element má své jméno a s každou vlastností elementu lze přímo pracovat, při vývoji aplikace i při jejím běhu. ArchestrA symboly nepřinášejí jen grafická vylepšení, ale také logická a funkční vylepšení vyplývající z použité objektové technologie. Intelligence ArchestrA symbolů je uložena ve skriptech, které jsou součástí symbolu. S okolím komunikují ArchestrA symboly prostřednictvím uživatelsky definovaných vlastností. Což jsou definice, které umožňují parametrizaci ArchestrA symbolu v aplikaci InTouch a umožňují také propojení na proměnné aplikace InTouch. Pro tvorbu ArchestrA symbolů slouží nové grafické

prostředí ArchestrA Symbol Editor. Toto prostředí nabízí daleko větší možnosti práce s grafikou než WindowMaker. Obsahuje sadu geometrických tvarů, jak základních, tak i složitějších tvarů, ze kterých se skládá vektorová grafika. Vlastnosti všech grafických objektů lze editovat. Vzhled nového prostředí si můžete prohlédnout níže.



Obr. 20 ArchestrA Symbol Editor

### Architektura ArchestrA

Představuje všezahrnující softwarové prostředí („průmyslový operační systém“). Zahrnuje v sobě nejdůležitější softwarové služby vyžadované aplikacemi průmyslové automatizace. Nejdůležitější jsou: robustní komunikace, zabezpečení, skriptování, centralizované alarmy a události, historizace, centrální nasazení, správa a diagnostika, centrální jednotné vývojové prostředí.

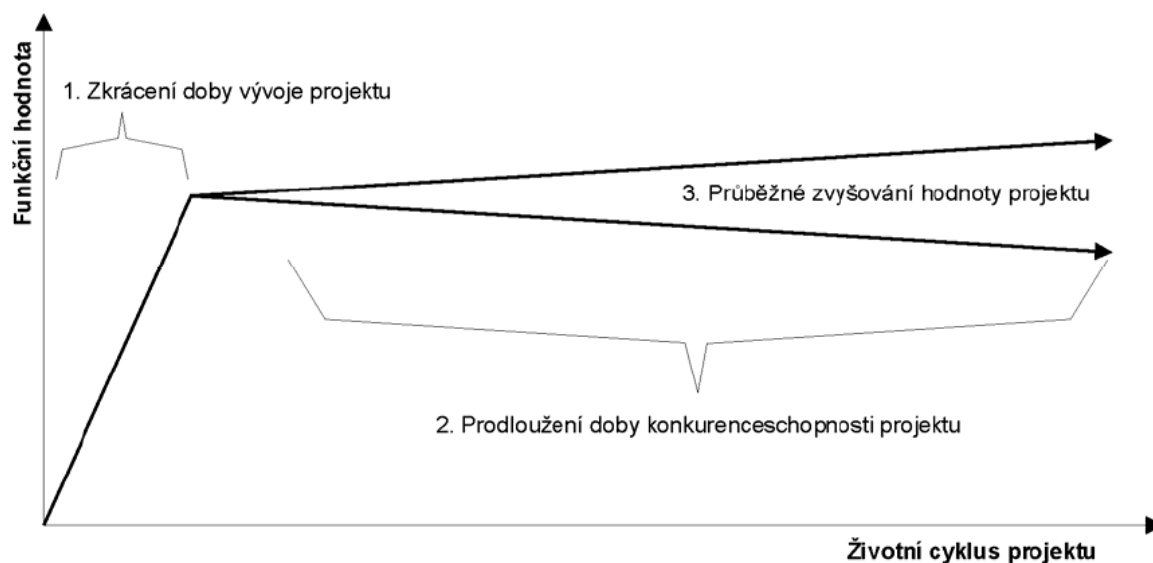
Infrastruktura ArchestrA poskytuje prostředky pro snadný a intuitivní vývoj modulárních průmyslových aplikací. Prostředí ArchestrA IDE generuje programový kód potřebný pro běh aplikace a umožňuje z jednoho místa nasazovat komponenty aplikace na cílové počítačové uzly. Tímto odstraňuje nutnost individuálně obcházet jednotlivé uzly a provádět místní instalaci.

Aplikace na bázi Wonderware System Platform 3.0 využívají možnosti objektové technologie. Tzn. všechny vlastnosti a funkčnosti (např. ventilu), jako jsou: grafický vzhled, animace I/O odkazy, konfigurace alarmů a historizace, skripty, definice

bezpečnostních úrovní a další funkčnosti; jsou vnořeny do ArchestrA aplikačních objektů. Z mateřských ArchestrA objektů je možné opakovaně odvozovat dceřiné objekty konkrétních zařízení. Toto velmi usnadňuje správu aplikace, protože veškerá logika a konfigurace vztahující se k danému objektu je uložena na jednom místě.

Architektura ArchestrA umožňuje opakovaně využít až 70% inženýrské práce z jednoho projektu v dalších projektech. Tím se podstatně zkracuje celkový čas vývoje automatizačních a informačních řešení. Také se snižují celkové náklady na vlastnictví automatizačních aplikací, protože architektura umožňuje jejich snadné vylepšování, čímž se prodlužují životní cykly projektů.

Dříve se vyvíjela každá aplikace vždy od počátku, a jakmile byl projekt předán uživateli, byl jeho životní cyklus časově omezen. Většinou časem docházelo k postupnému snižování konkurenceschopnosti dané aplikace a pro nové produkty bylo potřeba vytvořit nové projekty. Při využití technologie ArchestrA mohou být životní cykly projektů prodlužovány „téměř neomezeně“. Díky jednoduššímu vylepšování a rozšiřování může být zvýšena hodnota původní aplikace i doba její konkurenceschopnosti.



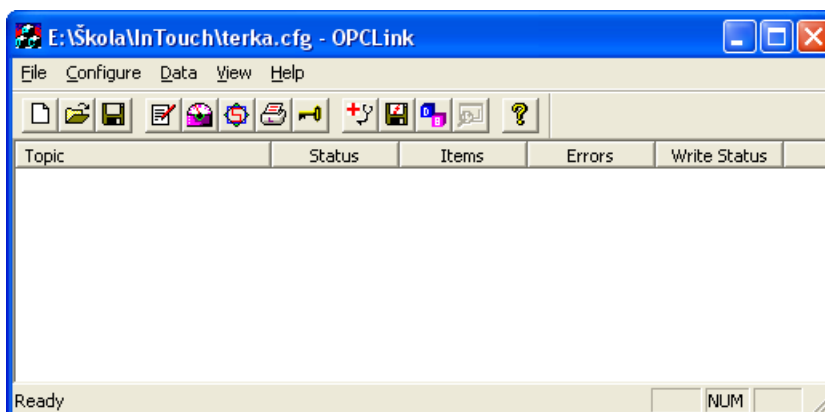
Obr. 21 Životní cyklus projektu při využití technologie ArchestrA [Pantek (CS) s.r.o.]

### Systémové požadavky

Wonderware Application Server 3.0 podporuje operační systémy Microsoft Windows XP SP2, Windows Server 2003 SP2 a Windows Vista (edice Enterprise, Business a Ultimate). MS Windows 2000 již není podporován. Pro konfigurační databázi Galaxy je vyžadována databáze Microsoft SQL Server 2005 SP2, která je však součástí licence systému Wonderware System Platform.

## 5.7 OPCLink 8.0

Aplikace Wonderware OPCLink slouží jako převodník komunikačních protokolů. Umožňuje ostatním aplikacím, které nepodporují standard OPC, připojit se k datům prostřednictvím lokálních nebo vzdálených OPC serverů. OPCLink se připojí k OPC serverům, převede klientské příkazy do protokolu OPC a zpátky předává data klientům s použitím DDE, FastDDE nebo SuiteLink.



Obr. 22 Aplikace Wonderware OPCLink 8.0

## 5.8 FactorySuite Gateway

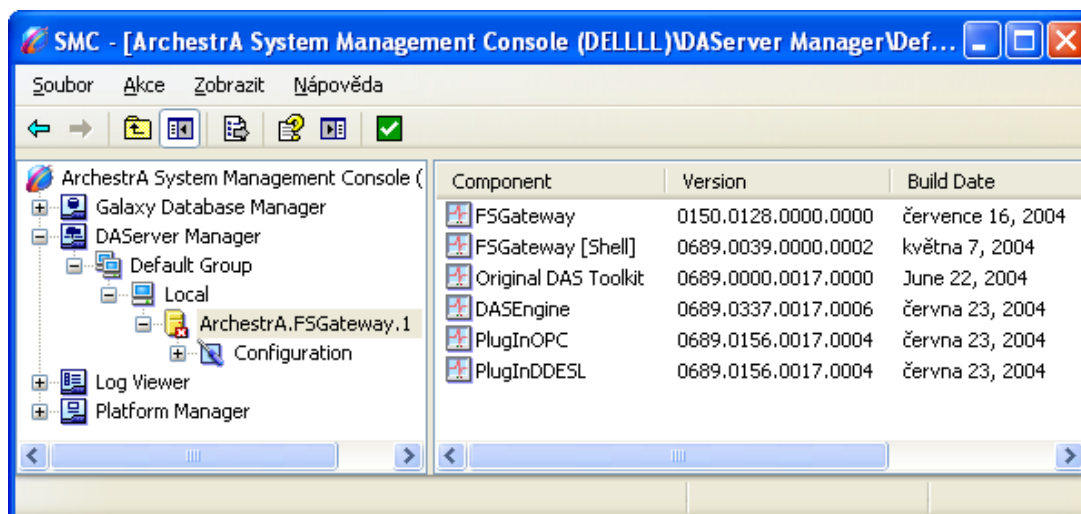
I tato aplikace slouží jako převodník komunikačních protokolů mezi zdrojem dat a klientem, který nepodporuje protokol OPC. FS Gateway byla vytvořena s ArchestrA DAS Toolkit. Může být použita k propojení klientů a zdrojů dat, které komunikují prostřednictvím rozdílných přístupových protokolů.

### Podporované komunikační protokoly:

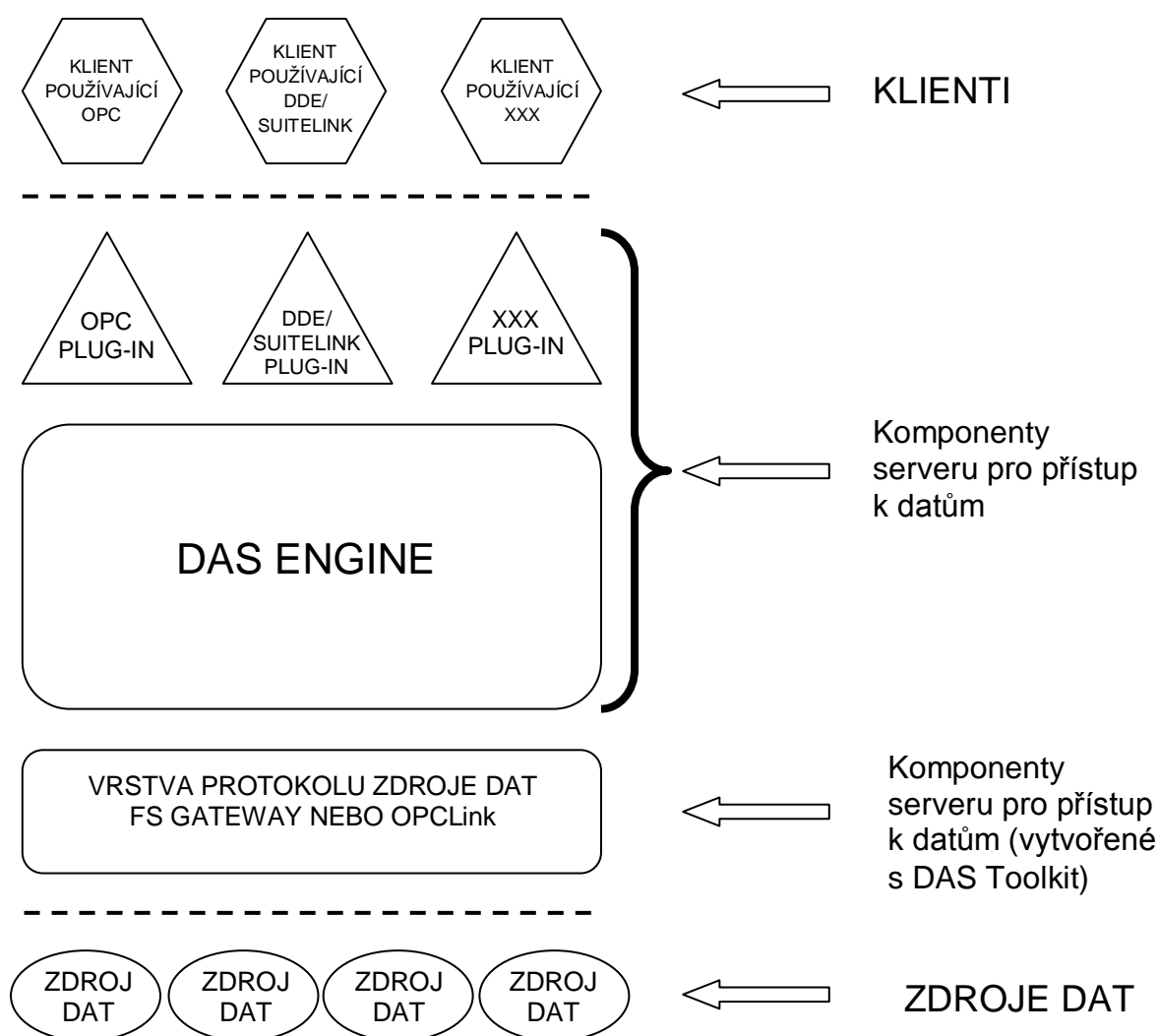
- OPC
- SuiteLink
- DDE
- FastDDE
- ArchestrA Message Exchange

Konfigurace FS Gateway se provádí přes DAServer Manager, který nalezneme v System Management Console (SMC). Před tím, než můžeme FS Gateway aktivovat, musíme stanovit komunikaci mezi zdrojem dat a klientem. Tuto hierarchii vytvoříme

přidáním uzlu do hierarchie FS Gateway. Jakmile je hierarchie vytvořena, může být každý zdroj dat konfigurován.



Obr. 23 Přístup k FS Gateway



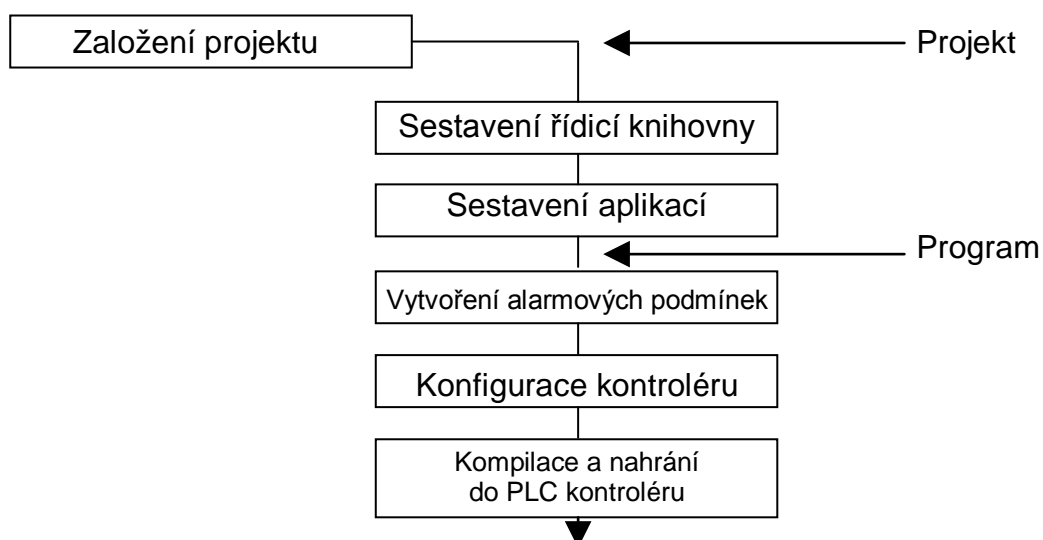
Obr. 24 Architektura FS Gateway / OPCLink

## 6 Demonstrační úloha a programovací prostředí

V laboratoři F204 je umístěna celá sestava úloh, jejíž součástí je i zařízení použité v této práci. Ke všem těmto úlohám bylo firmou ABB vytvořeno demo v prostředí Control Builder. Demonstrační projekt nese název Demo\_VSB a obsahuje celkem pět demo aplikací. Konkrétně se jedná o aplikace Coffe\_Machine\_SFC (automat na kávu), Door\_full\_automatic (otevírání a zavírání dveří), Pohon\_Zkouska (asynchronní motor), Test\_Cidlo (snímače) a poslední Trafic\_Lights\_SFC (řízení semaforu na křižovatce). Dále se budu podrobněji věnovat aplikaci Pohon\_Zkouska, na které si také přiblížíme uživatelské prostředí PLC Control Builderu a jeho editoru. Control Builder používá určitou programovou hierarchii, kterou je nutné si uvědomit. Tuto hierarchii popisuje následující seznam:

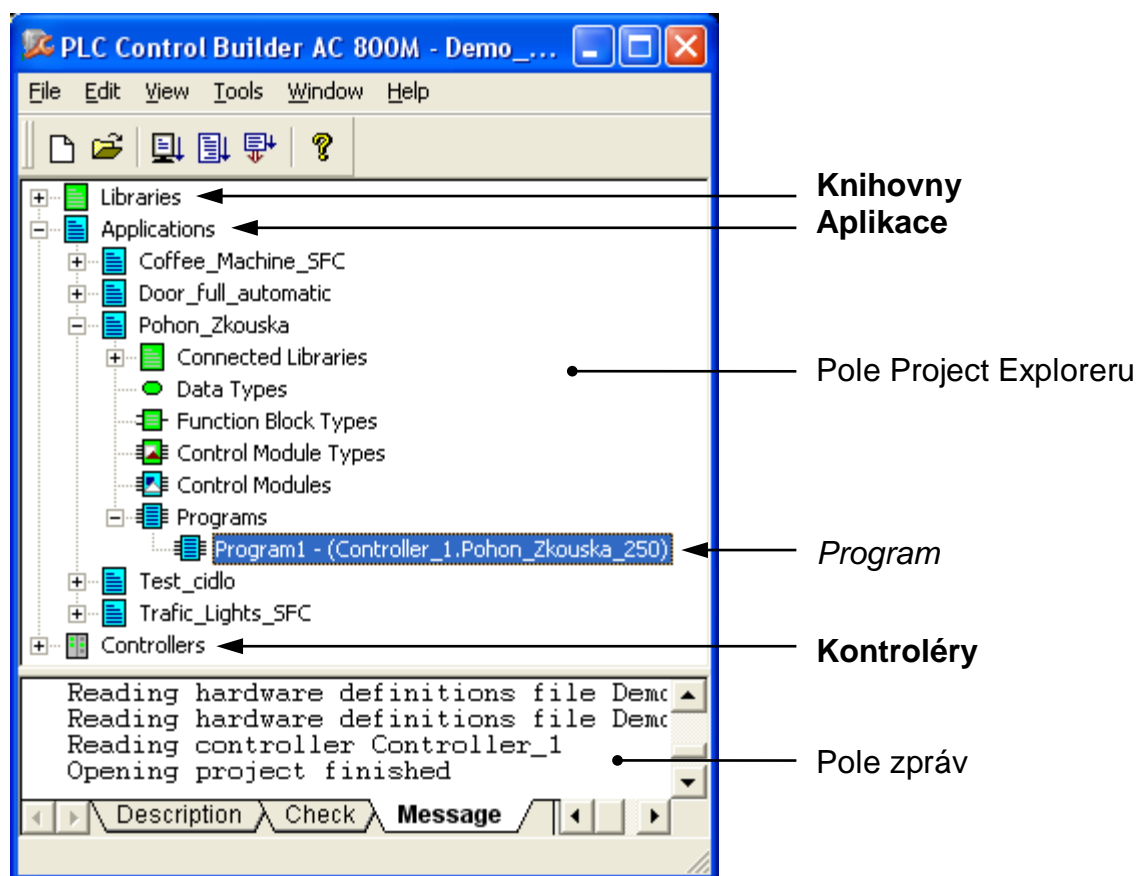
- **projekt** – nejvyšší úroveň softwarové jednotky, obsahuje konfigurační data pro knihovny, aplikace, připojený hardware atd. Seskupuje knihovny, aplikace a připojený hardware ve struktuře hierarchického stromu v Project Exploreru.
- **aplikace** – obsahuje programy a další objekty (datové typy, druhy funkčních bloků, typy řídicích modulů) které jsou v rámci aplikace použity.
- **program** – je připojen k procesu, který rozhoduje, jak často je program vykonáván. Je také možné připojit jednotlivé funkční bloky a řídicí moduly k různým procesům.

Sekvence níže znázorňuje kroky od vytvoření nového projektu až po jeho zavádění.



Obr. 25 Pořadí tvorby projektu a hierarchie mezi projektem a programy

Jádro uživatelského rozhraní PLC Control Builderu tvoří Project Explorer. Jedná se o prohlížeč, který se zobrazí hned při spuštění programu. Najednou může být otevřen jen jeden projekt. Vzhled tohoto prohlížeče si můžete prohlédnout na obrázku níže (Obr. 26). Takto vypadá po otevření projektu Demo\_VSB.



Obr. 26 Prostředí PLC Control Builderu (Project Explorer)

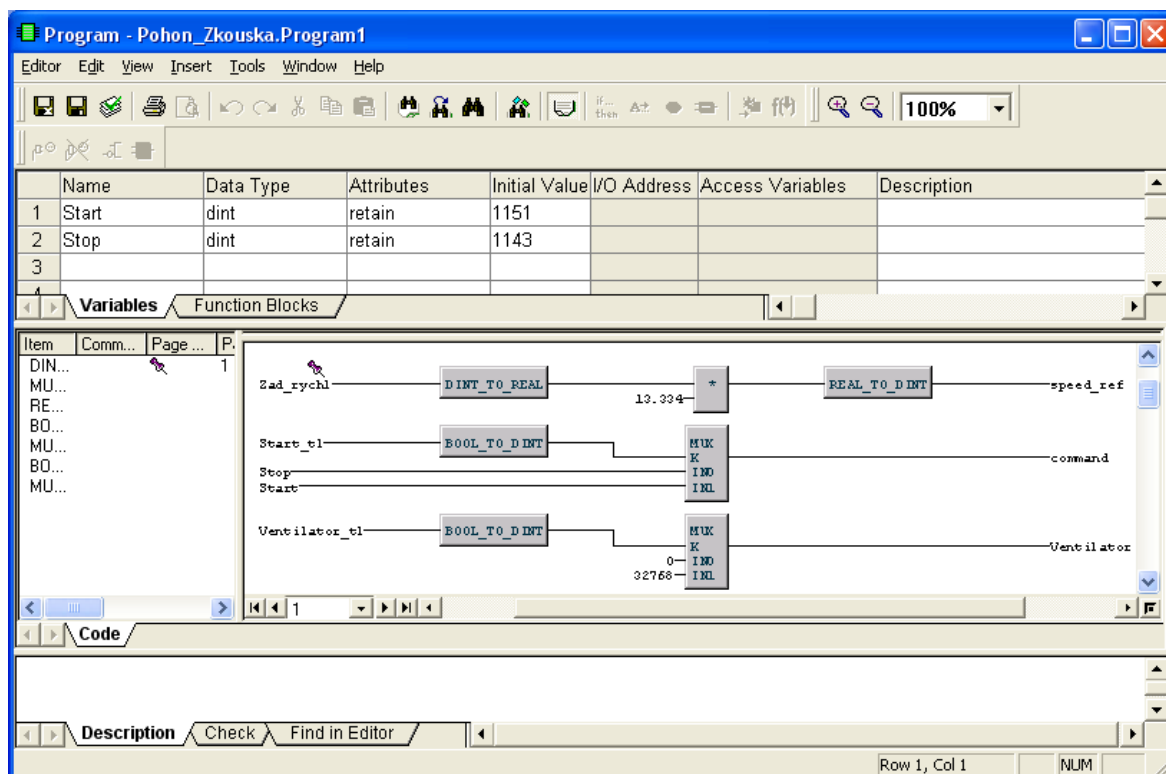
Všimněte si členění složek. Pole Project Exploreru obsahuje tři hlavní složky:

- Libraries (knihovny)
- Applications (aplikace)
- Controllers (kontroléry)

Control Builder obsahuje řadu editorů, které mohou být zpřístupněny z Project Exploreru. Pro přístup k těmto editorům stačí kliknout pravým tlačítkem myši na požadovaný objekt (může to být kontrolér, další hardwarová jednotka, aplikace, program nebo typ) a vybrat editor z kontextové nabídky. Editory slouží, mimo jiné, k deklaraci konstant a parametrů projektu, k deklaraci proměnných a jejich připojení k I/O kanálům.

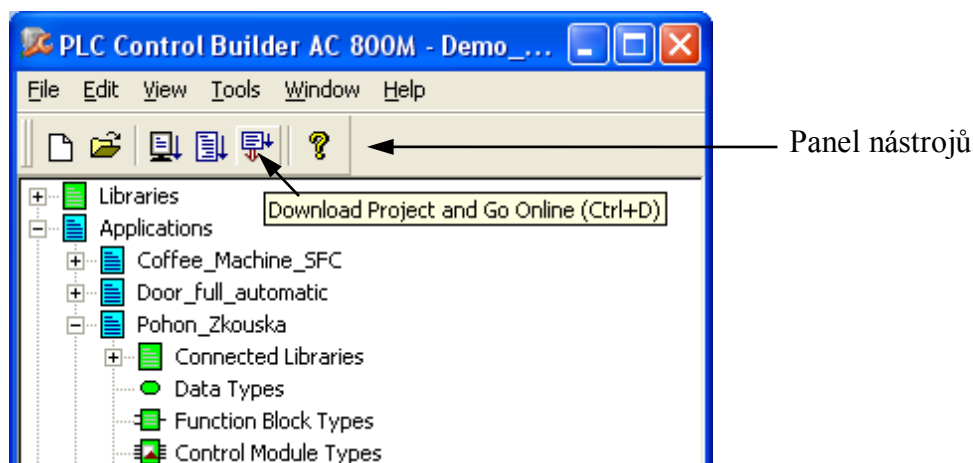


Obrázek níže (Obr. 27) ukazuje editor programu Pohon\_Zkouska.Program1. Tento program je napsán v jazyce FBD (funkční blokové schéma).



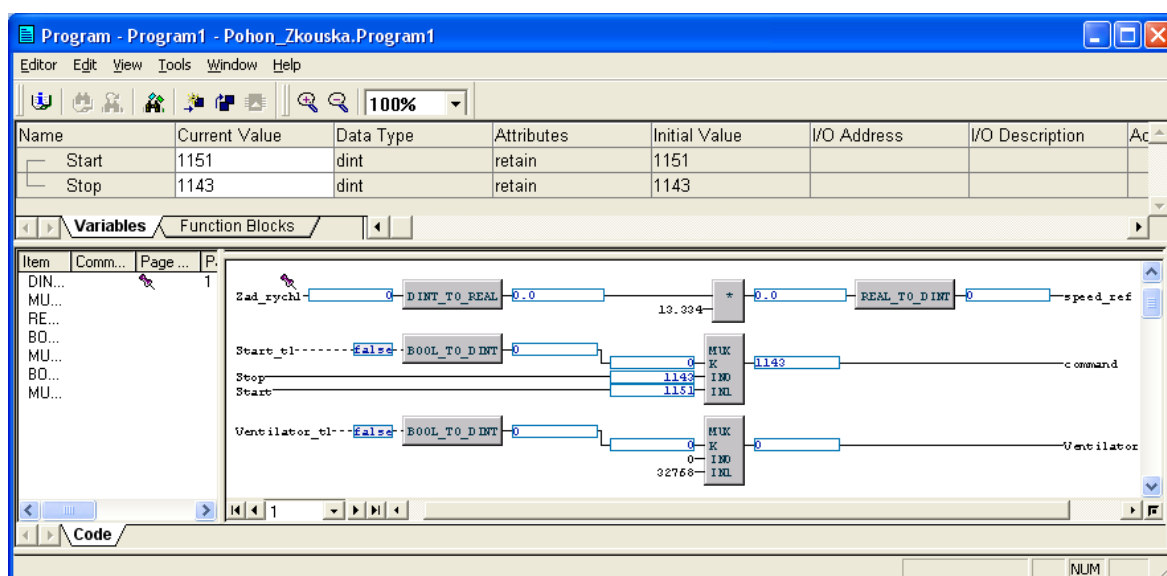
Obr. 27 Editor programu

Nahrání projektu do PLC se provádí pomocí ikon na panelu nástrojů. Control Builder nabízí tři možnosti spuštění projektu (3 ikony). První možností je testovací režim, při kterém se projekt spustí jen na počítači bez kopírování do PLC. Další ikona slouží pouze pro přechod do online režimu (v případě kdy už máme projekt v PLC nahrán). Stiskem poslední ikony provedeme nakopírování projektu do PLC a zároveň se přepneme do online režimu (Obr. 28).



Obr. 28 Nahrání projektu do PLC a jeho spuštění

Po nakopírování projektu do PLC a přechodu do online režimu můžeme vybraný program také otevřít v editoru (dvojklikem na název programu). Otevře se okno prohlížeče, ve kterém opět vidíme vybraný program, avšak tentokrát v online režimu. Je tedy možné vyzkoušet jeho funkce a sledovat jeho chování. Na obrázku níže si můžete prohlédnout online režim programu určeného pro pohon. V tomto režimu je možné měnit vstupní parametry (zadat rychlost, spustit/zastavit motor a zapnout/vypnout ventilátor). Ihned po nastavení se změny projeví na výstupu.

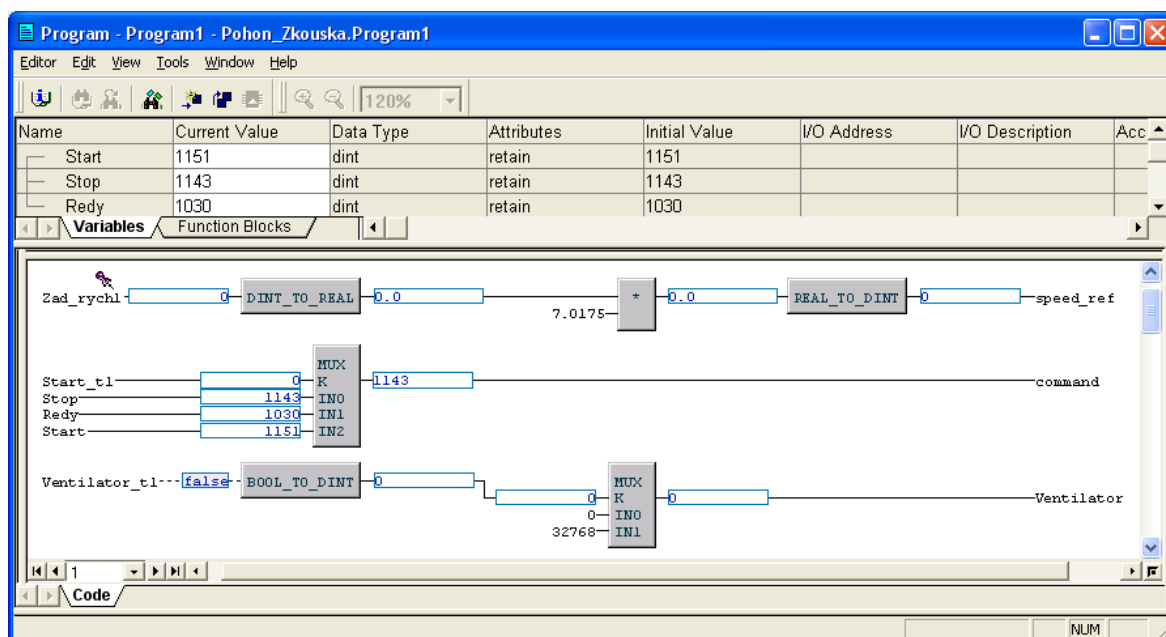


Obr. 29 Online režim programu

### Oprava demo programu

Tento program (Pohon\_Zkouska) však obsahoval drobnou chybu, která většinou způsobovala nefunkčnost aplikace. Konkrétně při rozběhu motoru. Problém spočíval v absenci řídicího slova 1030 (příprava ke spuštění), které je nutné měniči před startem motoru poslat. Důsledkem toho nebylo možné za běžných okolností motor spustit. Měnič sice reagoval na změnu referenčních otáček, avšak k rozběhu motoru nedošlo. Program jsem tedy poopravil a doplnil proměnnou („Redy“), jež dokáže řídicí slovo 1030 měniči poslat. Následně jsem program otestoval. Vše fungovalo již bez problémů. Upravený program spuštěný v online režimu můžete vidět na následujícím obrázku.

Druhá chyba byla v převodu otáček. Tudíž pokud uživatel zadal určité referenční otáčky, tak na měniči se nastavily otáčky jiné. Problém byl ve špatně vypočtené konstantě pro násobení.



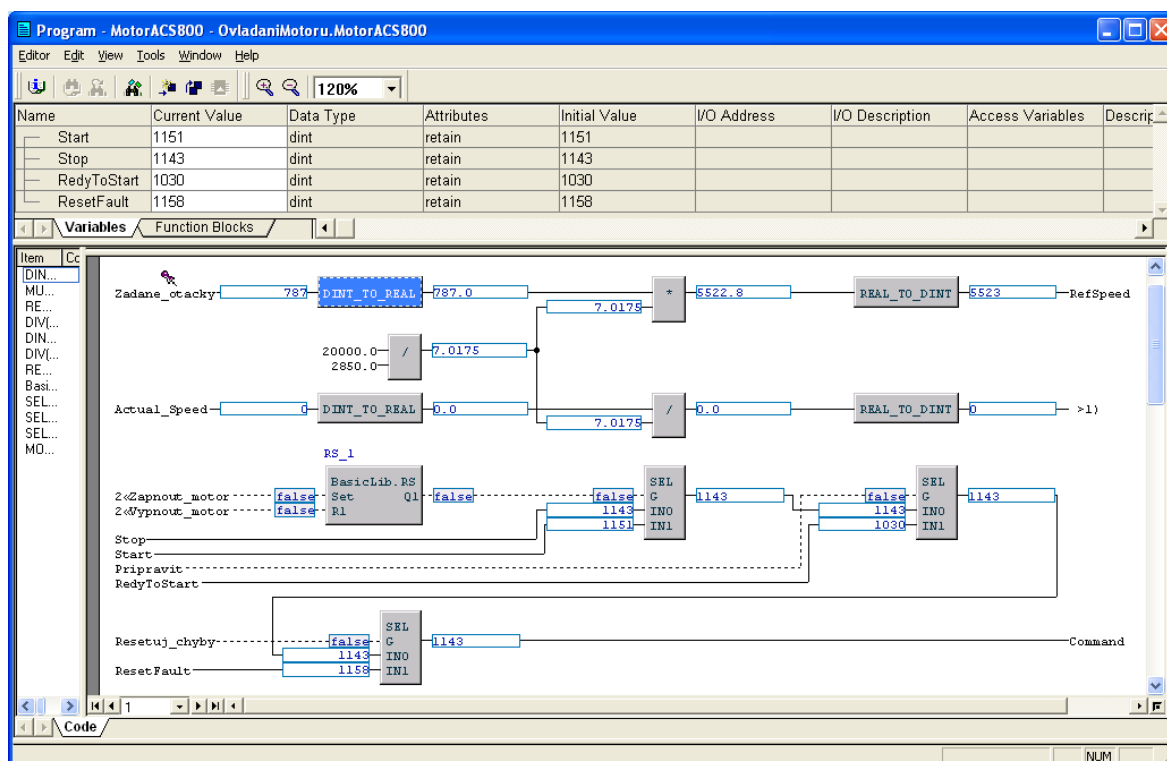
Obr. 30 Upravený demo program (Pohon\_Zkouska)

## 6.1 Řídicí program pro PLC

Za účelem vizualizace jsem pro PLC napsal nový program, pomocí kterého bude celá úloha řízena. Zvolil jsem programovací jazyk FBD. Výsledný program má tyto funkce: umožňuje zadávání referenčních otáček, rozběh a zastavení motoru, již dříve zmíněnou „přípravu na start“ a také možnost resetovat případně vzniklé chyby měniče. Dalším úkolem programu je předávat data z měniče dále do aplikace InTouch a naopak.

Při zadávání referenčních otáček je nutné provádět přepočet. Měnič totiž s otáčkami nepracuje v běžných hodnotách, ale v tzv. poměrových jednotkách (0 – 100%). Měnič pracuje s rozsahem 0 – 20000. Přičemž 0 = 0% a 20000 = 100%. Což znamená, že podle maximálních otáček použitého motoru musíme rozsah přepočítat. Vypočteme tak hodnotu, kterou musíme zadávané otáčky násobit. V mém případě má motor jmenovité otáčky 2850ot/min. Tudíž se konstanta pro násobení vypočte vztahem  $20000 / 2850 = 7,0175$ .

Program spuštěný v online režimu si můžete prohlédnout na následujícím obrázku. V horní části okna prohlížeče jsou definována řídicí slova pro měnič. Je třeba si uvědomit, že samotné použití slov určených pro start (1151) a stop (1143) motoru nestačí.



Obr. 31 Vlastní program pro PLC

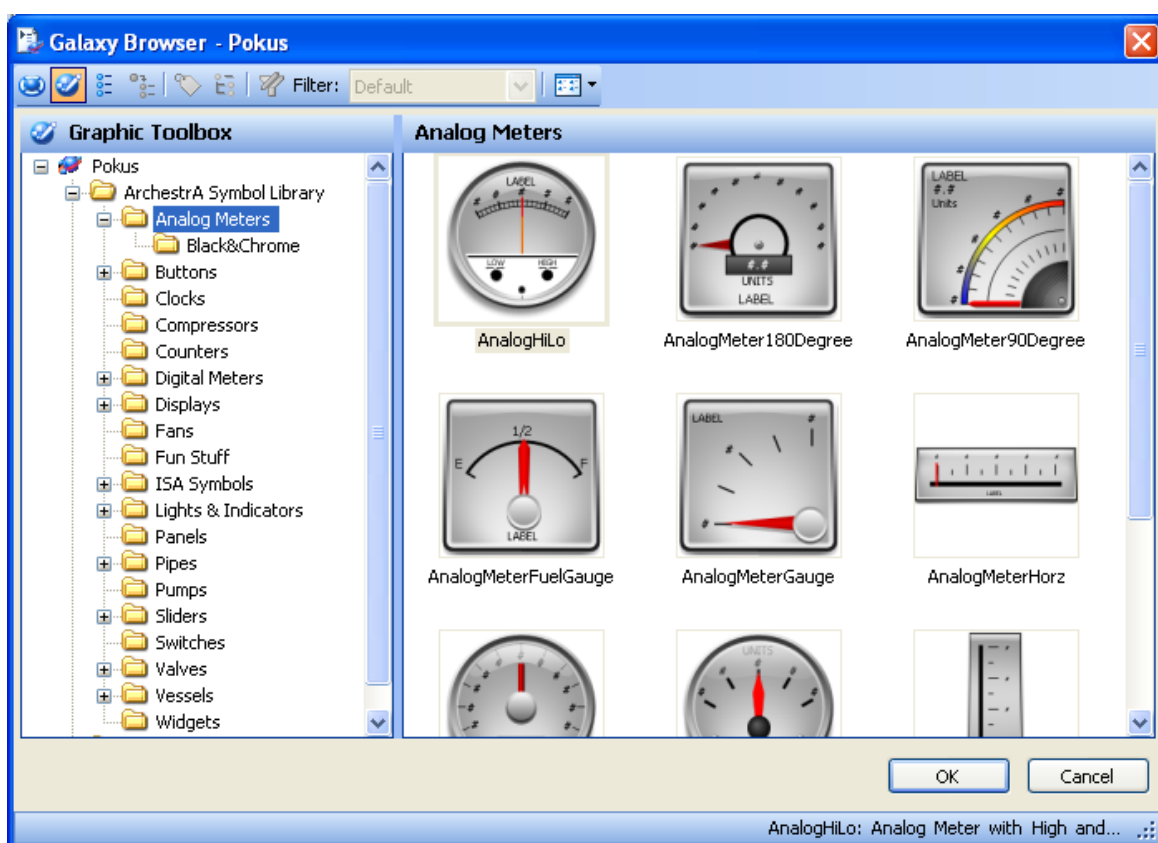
## 6.2 Aplikace pro vizualizaci v InTouch

Pro vizualizaci úlohy slouží aplikace vytvořená v programovém prostředí InTouch. Zvolil jsem tvorbu pomocí integrovaného vývojového prostředí ArchestrA IDE. Tudíž tato aplikace spadá do kategorie InTouch aplikací spravovaných z IDE. Jak již bylo zmíněno, tyto aplikace umožňují využití všech nových možností technologie ArchestrA. V mém případě se jednalo zejména o využití ArchestrA grafiky, respektive knihovny ArchestrA symbolů.

Prostředí ArchestrA IDE obsahuje rozsáhlou knihovnu stovek již hotových ArchestrA symbolů, kterou lze také rozšiřovat o vlastní ArchestrA symboly. Symboly najdeme v aplikaci pod ikonou „Embed ArchestrA Graphic“. Knihovna je členěna do stromové struktury, v níž se nacházejí skupiny (složky) prvků, např. Analog Meters atd.

Funkčnost programu InTouch ve vývojovém prostředí ArchestrA IDE je ovládána dvěma automatizačními objekty:

- objekt InTouchViewApp představuje aplikaci InTouch při navrhování a běhu programu, pomocí tohoto objektu architektura ArchestrA spravuje InTouch aplikace
- objekt ViewEngine kontroluje běh aplikace InTouch v cílových uzlech v databázi Galaxy



Obr. 32 Knihovna již hotových ArchestrA symbolů

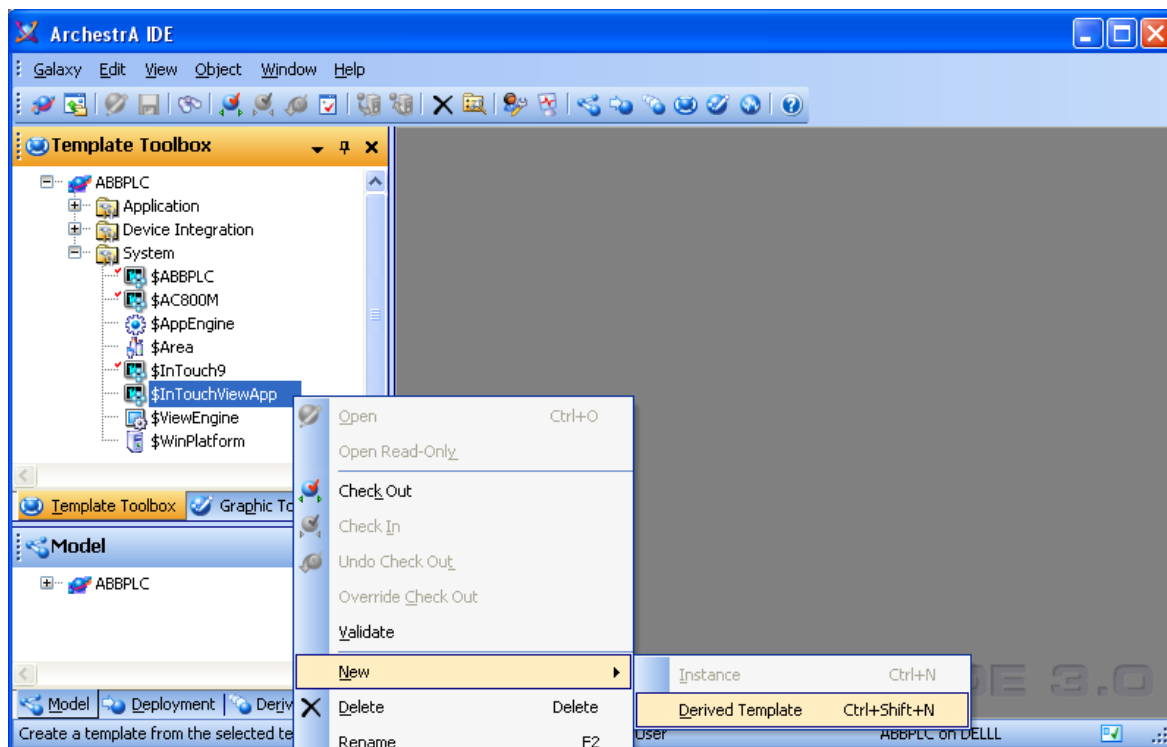
### Postup při tvorbě aplikace

Tvorba aplikací v IDE, respektive jejich zakládání, se poněkud liší od klasických InTouch aplikací. Proto zde stručně nastíním základní kroky, které musí začínající uživatel před samotnou tvorbou vizualizace podniknout.

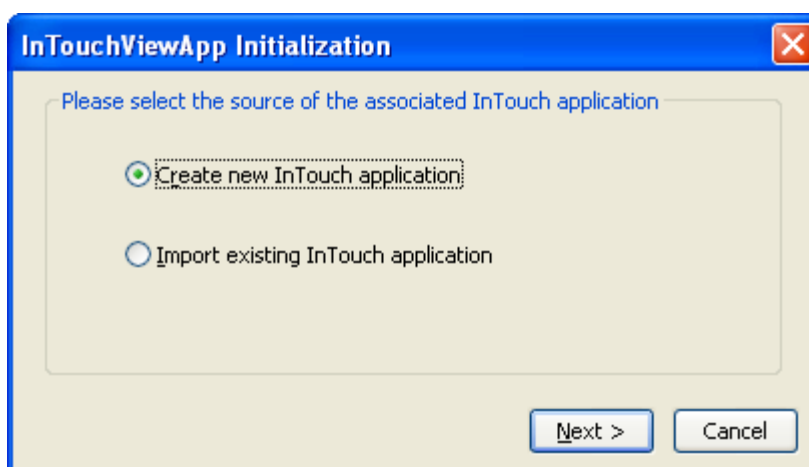
Ještě před založením projektu, hned při spuštění prostředí ArchestrA IDE je nutné vytvořit připojení na tzv. Galaxy. Jedná se v podstatě o připojení k aplikačnímu serveru.

Pokud je Galaxy vytvořena, pak se k ní můžeme připojit a pustit se do založení nového projektu (aplikace).

Prvním krokem při založení InTouch aplikace spravované z IDE je vytvoření nové InTouchViewApp šablony. To provedeme tím, že od stávající šablony odvodíme šablonu novou. Poté co tuto šablonu vytvoříme, můžeme ji asociovat s InTouch aplikací. Nabízí se dvě možnosti – vytvoření nové InTouch aplikace, nebo importování klasické InTouch aplikace. Postup založení aplikace popisují následující obrázky.



Obr. 33 Vytvoření nového objektu InTouchViewApp

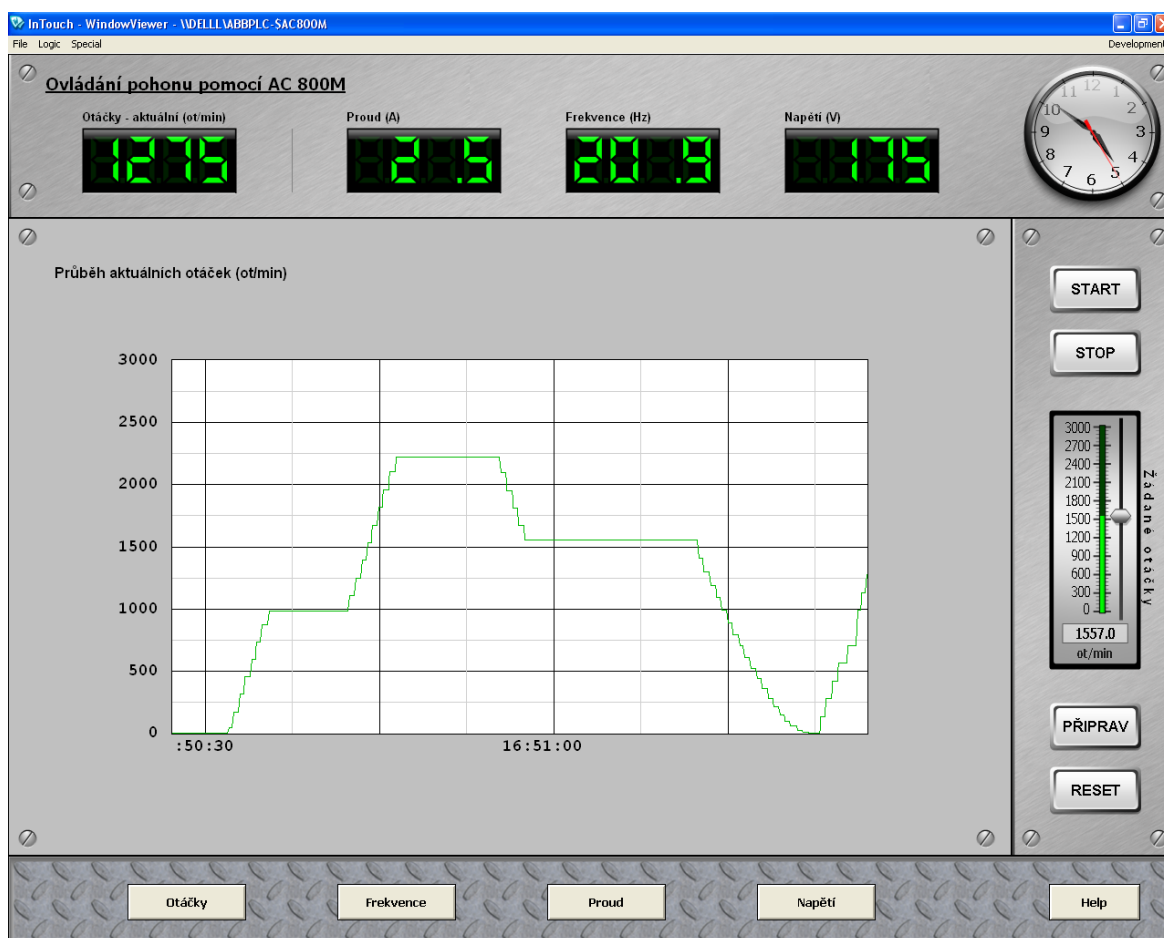


Obr. 34 Asociace nově vytvořené šablony s InTouch aplikací

Založenou aplikaci můžeme upravit spuštěním programu WindowMaker známého již z předchozích verzí InTouch. WindowMaker spustíme dvojklikem na odpovídající šablonu InTouchWiewApp. Prostředí se od předchozích verzí moc neliší, až na podporu nové grafiky v podobě knihovny již hotových ArchestrA symbolů.

V navržené aplikaci jsem využil zejména symbolů pro zobrazování hodnot monitorovaných veličin. Dále jsem použil posuvník pro nastavení otáček motoru a nová ovládací tlačítka. Aplikace je doplněna také o hodiny.

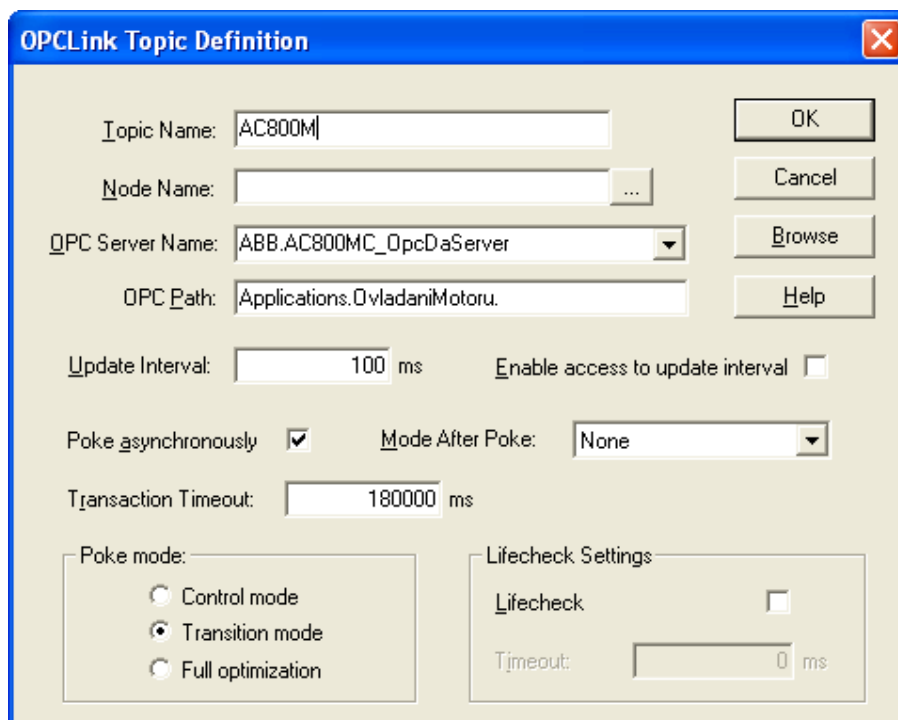
V horní části obrazovky jsou umístěny displeje a jeden analogový ukazatel pro zobrazení aktuálních hodnot monitorovaných veličin. Na pravém panelu leží tlačítka pro zapnutí a vypnutí motoru, pod nimi posuvník pro nastavení požadovaných otáček motoru. Další dvě tlačítka pod posuvníkem slouží k přípravě měniče ke startu, anebo resetování případných poruchových hlášení měniče. Největší plocha je věnována grafu, který vykresluje aktuální průběh zvolené veličiny. Tlačítka umístěná dole na obrazovce slouží pro přepínání prostředního okna a pro zobrazení jednoduché nápovědy týkající se použití tlačítek PŘIPRAV a RESET. Pro přepínání oken musela být použita starší tlačítka Nová tlačítka z aplikačního serveru totiž přepínání oken (v InTouch 10.0) neumožňují.



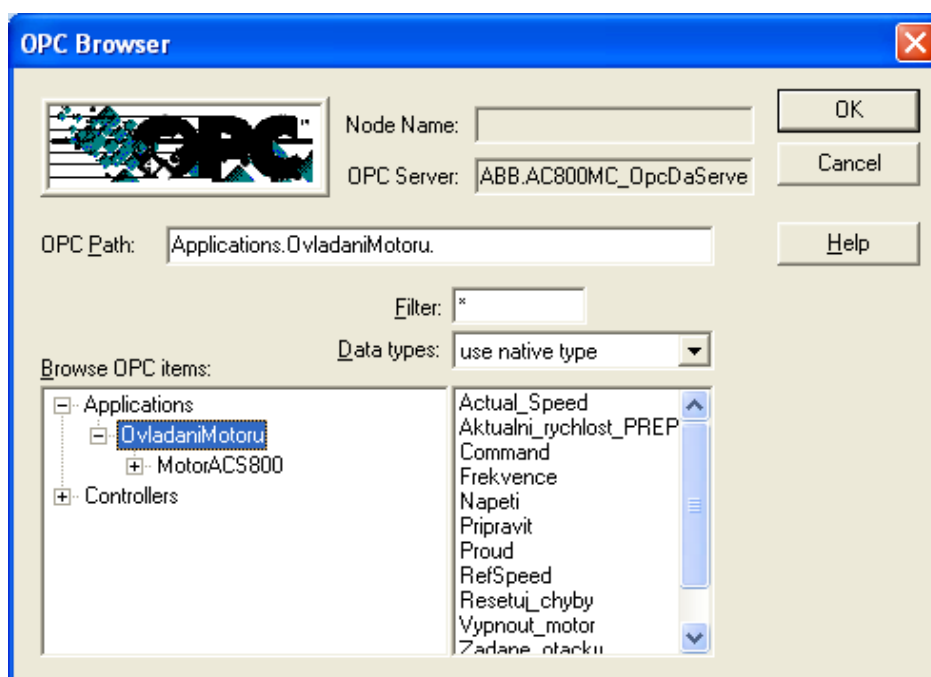
Obr. 35 Vzhled vytvořené aplikace pro vizualizaci

### 6.3 Nastavení komunikace

Jelikož InTouch nepracuje přímo s OPC, bylo nutné pro komunikaci použít další aplikaci jako převodníku. Zvolil jsem aplikaci OPCLink. Po její instalaci je nutné vytvořit nový topik a nakonfigurovat připojení k OPC serveru. Dále je nutné zadat adresu (OPC Patch). Popřípadě jméno uzlu, pokud je počítač v síti.



Obr. 36 Konfigurace topiku

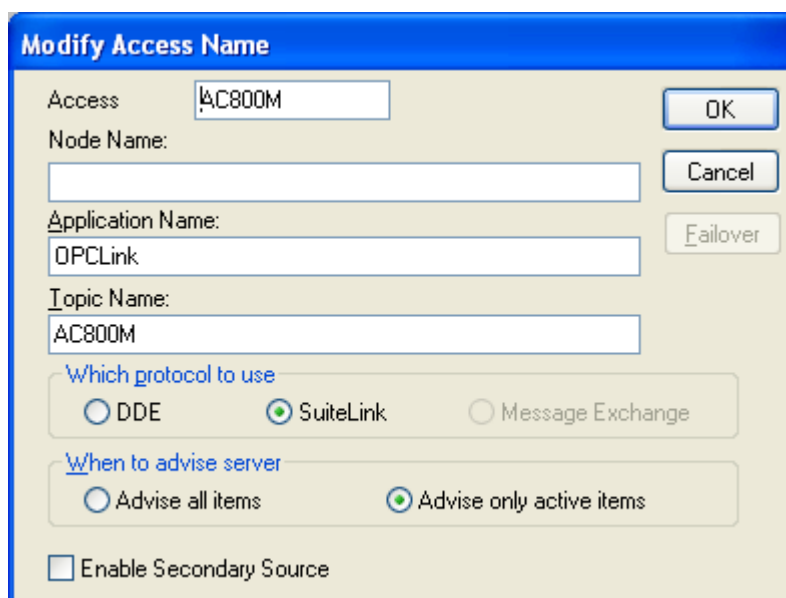


Obr. 37 OPC Browser



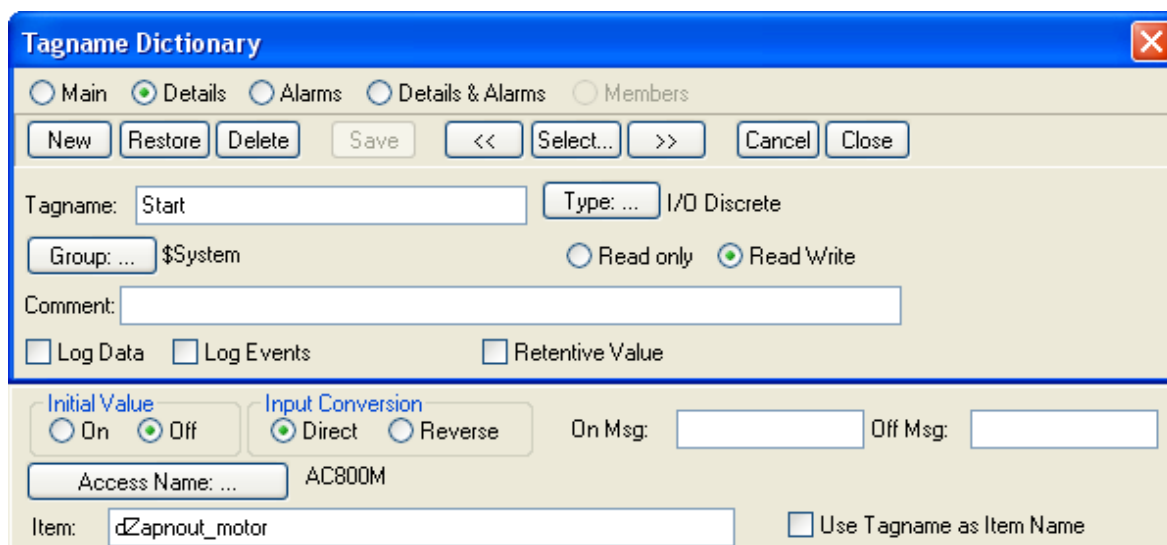
Pokud si s OPC cestou nejste jisti je nejlépe využít prohlížeč OPC (OPC Browser). Tato funkce najde veškeré OPC servery na vašem počítači a po výběru konkrétního serveru můžete přímo listovat v nabízených proměnných. Program pak za vás cestu automaticky vyplní.

V programu InTouch musíme samozřejmě také toto spojení nastavit. To znamená, při přiřazení proměnné k danému objektu musíme nejprve vytvořit přístupové jméno Access name (stejně jako při definici topiku v OPCLink). A následně provést konfiguraci připojení.



Obr. 38 Definice připojení proměnné v InTouch

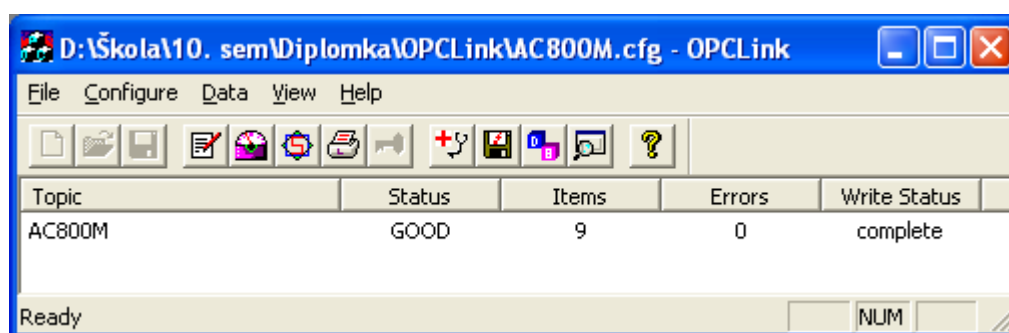
Pokud máme připojení nadefinované, přistoupíme k vytvoření proměnné. K tomu slouží Tagname Dictionary. V zásadě se bude jednat o I/O proměnné.



Obr. 39 Vytvoření I/O proměnné a připojení na komunikační kanál

Tagname si můžeme zvolit libovolně, tento název slouží pro uložení a identifikaci proměnné v InTouch aplikaci. To znamená, že pod tímto jménem se na danou proměnnou budeme odkazovat z jednotlivých objektů, ve kterých ji budeme chtít používat. Následně musíme pro I/O proměnné vybrat Access Name (přístupové jméno). Protože Access Name odkazuje na celou skupinu proměnných, musíme po jeho přiřazení ještě definovat, kterou proměnnou chceme číst. To provedeme zápisem položky Item. Do ní vepíšeme přesný název proměnné, kterou z OPC serveru požadujeme. Před název proměnné je nutné vložit jeden znak (písmeno) které udává, o jaký typ proměnné se jedná.

To že nám komunikace funguje, poznáme tak, že při spuštění aplikace využívající definovaný topik (v našem případě vizualizace pomocí InTouch) se topik zobrazí v okně OPCLink. Můžeme tak sledovat kolik proměnných je použito při komunikaci a také vidíme jejich stav. Při běhu aplikace však není možné měnit parametry topiku. Pokud chceme něco změnit, je nutné nejprve aplikaci vypnout.



Obr. 40 OPCLink při běhu vizualizace

## **7 Návrh směru dalšího řešení**

Směr dalšího řešení by mohl spočívat například v návrhu a následné realizaci připojení zátěže nebo brzdy k motoru. Bylo by tak možné monitorovat motor při různých stavech zatížení a sledovat chování celé soustavy.

Další možností by mohlo být rozšíření aplikace pro vizualizaci. Například doplnění modulů statistiky (SPC pro kvalitu řízení atd.). Celá sestava by mohla být také doplněna připojením k ovládacímu panelu (Process Panel).

## 8 Závěr

V rámci mé práce jsem se nejprve seznámil s problematikou asynchronních motorů. Základní rozdělení těchto strojů, jejich hlavní části, princip činnosti a také způsoby jejich řízení (regulace otáček). Věnoval jsem se také nabídce motorů vyráběných firmou ABB.

Dalším krokem bylo seznámení s demonstrační úlohou od firmy ABB. Tato sestava je umístěna ve školní laboratoři F204. Jedná se o asynchronní motor s kotvou na krátko řízený frekvenčním měničem ACS 800. Sestava se připojuje přes frekvenční měnič pomocí optického kabelu ke stolnímu počítači, pomocí kterého můžeme soustavu řídit a provádět různé měření (monitorování signálů). Počítač musel být pro tyto účely vybaven adaptérem PCI/PCMCIA, který umožnil osazení PCMCIA karty do slotu PCI. Po zapojení následovalo testování komunikace mezi PC a měničem. Pro ovládání měniče z PC je určen softwarový nástroj DriveWindow. Tento software umožňuje nejen ovládání měniče, ale také nastavení parametrů jeho firmware a monitoring pohonu.

K demonstrační úloze bylo následně připojeno také PLC (AC 800M). Tudíž dalším postupem bylo vyzkoušení komunikace mezi počítačem a PLC. Za tímto účelem jsem se seznámil s demonstračním programem, který jsem následně vyzkoušel nahrát do PLC. Jelikož PLC původně nebylo spojené dále s frekvenčním měničem, musel jsem vyřešit problematiku tohoto propojení. Původně se předpokládalo propojení PLC s měničem přes rozhraní optické ModuleBus. Nakonec jsem ale propojení realizoval pomocí sběrnice ProfiBus. Přes optický kabel je již připojen frekvenční měnič k PC. Toto propojení slouží k ovládání měniče použitím softwarového nástroje DriveWindow. Jelikož se jedná v podstatě o dvě nezávislé úlohy, je zapotřebí obou propojení. Z důvodu vyzkoušení obou sběrnic jsem nakonec zvolil pro připojení k PLC rozhraní ProfiBus.

Po seznámení s hardwarovou částí projektu následovalo seznámení s použitým programovým vybavením. Vyzkoušel jsem si ovládání celé řady programů. Jednalo se o programy sloužící k obsluze a programování PLC nebo frekvenčního měniče a programy potřebné následně k vizualizaci úlohy. Mezi tyto programy patří DriveWindow 2.12, PLC Control Builder AC 800M, OPC Server for AC 800M (verze 4.1), SoftController 4.1, InTouch 10.0, Wonderware Application Server 3.0 a prostředí ArchestrA IDE, OPCLink 8.0 a FS Gateway. V rámci své práce jsem stručně popsal každý z nich.

V další fázi jsem začal s úvodem do programování v nástroji PLC Control Builder. K čemuž jsem použil demonstrační úlohu napsanou v jazyce FBD (funkční blokové schéma).

Seznámil jsem se s principy programování PLC a programovacími jazyky k tomu určených. Zejména s technologií od firmy ABB. Upravil jsem demonstrační program, který byl k úloze dodán. A poté vytvořil vlastních aplikací, jež následně používám pro komunikaci s vizualizací.

Posledním bodem byla tvorba vizualizační aplikace. Zvolil jsem, v té době poslední verzi, InTouch 10.0. Pro vytvoření aplikace jsem použil nové vývojové prostředí ArchestrA IDE, tak abych mohl využít potenciálu nové ArchestrA grafiky, kterou tato architektura přináší. K monitorování jsem zvolil celkem čtyři proměnné – napětí, proud, frekvenci a otáčky motoru. Otáčky motoru bude možné z aplikace i nastavit. Vzájemnou komunikaci aplikací prostřednictvím OPC protokolu zprostředkovává aplikace OPCLink.

Výsledná úloha by měla sloužit pro ukázky při výuce jako příklad představující reálné využití v praxi.

## Použitá literatura

ABB, *Manuál pro obsluhu demonstračního zařízení dodaného společností ABB na VŠB – TU*. [uloženo v laboratoři]

ABB, *Basic Control Software Introduction and Configuration*.

[3BSE035980R4001\_CIO\_BasicSW.pdf]

ABB, *Industrial Compact Control Builder Version 4.1, Getting Started Introduction and Installation*.

[3BSE039838R201\_en\_Compact\_Control\_Builder\_SV\_4.1\_Getting\_Started[1].pdf]

FARANA, R., SMUTNÝ, L., VÍTEČEK, A., VÍTEČKOVÁ, M. 2004. *Zpracování závěrečných prací z oblasti automatizace a informatiky*. 1.vyd. Ostrava: VŠB-TU, 2004. 116 s. ISBN 80-248-0557-X.

CENDELÍN, J. *Historie programovatelných automatů a jejich současné efektivní použití*.

[online]. Plzeň: Západočeská universita. [cit. 2008-10-06]. Dostupné z www:

<<http://www.odbornecasopisy.cz/automa/2003/au060306.htm>>

Help k programu Factory Suite 2000, InTouch 9,5 [CD ROM].

Landryová, L. - Pawelek, M. - Konečný, M.: *Návrh procesních systémů. Programové systémy SCADA/MMI*. 1. vyd. Ostrava, KAKI 1996, 96 s. ISBN 80-02-01100-7.

Wonderware 1997. *Wonderware's Communication Concept FactorySuite™2000*.

[online]. Irvine, CA: Wonderware, corp., 1997. 12 s. Dostupné z www: < URL:

[http://www.klgssystem.com/product-services/automation/pdf/FS2K\\_Communication\\_Concepts.pdf](http://www.klgssystem.com/product-services/automation/pdf/FS2K_Communication_Concepts.pdf) >

ABB 2006. *DriveWindow 2 User Manual*. [User\_Manual\_EN.pdf]

ABB 2007. *ACS800 Hardware Manual*. [en\_800\_01\_hw\_G\_sccres.pdf]

ABB 2005. ACS800, *Programovací manuál, ACS800 Standardní aplikační program 7.x*.

[ACS800\_fw\_cz\_.pdf]

ABB 2007. *ACS800, Firmware Manual, System Control Program 7.x*.

[EN\_ACS800\_SystemControlProgram\_FW\_d.pdf]

ABB 2006. *Low Voltage General Purpose Motors*. [General\_purpose\_motors.pdf]

ABB 2006. *Low and High Voltage Process Performance Motors*.

[Process\_performance\_motors.pdf]

Wikipedia. *Asynchronní motor*. [online]. Dostupné z www:

<[http://cs.wikipedia.org/wiki/Asynchronn%C3%AD\\_motor](http://cs.wikipedia.org/wiki/Asynchronn%C3%AD_motor)>

*Elektrické pohony*. [online]. Dostupné z www: <[http://e-](http://e-automatizace.vsb.cz/ebooks/ridici_systemy_akcni_cleny/Akc_el.html)

[automatizace.vsb.cz/ebooks/ridici\\_systemy\\_akcni\\_cleny/Akc\\_el.html](http://e-automatizace.vsb.cz/ebooks/ridici_systemy_akcni_cleny/Akc_el.html)>

Kuchař, Martin. Štěpanec, Libor. *Přímé řízení asynchronních motorů – Takahashiho metoda*. [online]. Automatizace, 2006. Dostupné z www:

<<http://www.automatizace.cz/article.php?a=1138>>

MatrikonOPC. *OPC Tutorial*. [online]. Edmonton, Alberta : MatrikonOPC. Dostupné z

www: < URL: [http://www.matrikonopc.com/training/opc-multimedia-tutorial/opc\\_tutorial\\_printable\\_version.pdf](http://www.matrikonopc.com/training/opc-multimedia-tutorial/opc_tutorial_printable_version.pdf) >

ABB 2005B. *OPC Server for AC 800M – Installation and Configuration*. [online].

Västerås (Sweden): Automation Technology Products, Květen 2005 [cit. 2007-08-04].

Číslo dokumentu: 3BSE035983R4101. < URL: [3BSE035983R4101\\_CIO\\_OPCTServ.pdf](#) >

ABB 2002B. *AC 800M Controller – The modular and scalable process*

*controller*. [online]. Västerås (Sweden): Automation Technology Products, 2002 [cit.

2006-04-03]. Číslo dokumentu: 3BSE 023 957 R0201. Dostupné z www: < URL: [AC+800M.pdf](#) >

Zezulka, F., Bradáč, Z., Fiedler, P., Kučera, P., Štohl, R. *Programovatelné automaty*.

Brno, 2003. [Programovatelne\_automaty-Skripta\_FEKT\_VUT\_Brno.pdf]. Fakulta

elektrotechniky a komunikačních technologií, VUT v Brně. Dostupné z www:

<[http://www.advant.cz/Dokumentace/Programovatelne\\_automaty/Programovatelne\\_automaty-Skripta\\_FEKT\\_VUT\\_Brno.pdf](http://www.advant.cz/Dokumentace/Programovatelne_automaty/Programovatelne_automaty-Skripta_FEKT_VUT_Brno.pdf)>

PROFIBUS Nutzorganisation e. V. *PROFIBUS Technologi and Application*. 2002.

[4002\_vOctober2002-English.pdf]. Dostupné z www:

<[http://www.automation.siemens.com/download/internet/cache/3/1370385/pub/de/4002\\_vOctober2002-English.pdf](http://www.automation.siemens.com/download/internet/cache/3/1370385/pub/de/4002_vOctober2002-English.pdf)>

PROFIBUS Nutzorganisation e. V. *Profibus RS 485-IS User and Installation Guideline*.

[RS485-IS\_User\_Inst\_2262\_V11\_jun03.pdf]

ABB, 2005. *PROFIBUS DP Adapter Module RPBA-01 User's Manual*.

[EN\_RPBA01\_UM\_F\_A4.pdf]

ABB, 2002. *Pulse Encoder Interface Module RTAC-01 User's Manual*.

[en\_rtac\_um\_a.pdf]

Osadník, P. *Modelování objektů průmyslových zařízení v prostředí průmyslového aplikačního serveru*, Ostrava: katedra automatizační techniky a řízení, VŠB-TU Ostrava, 2008. 69 s. Diplomová práce, vedoucí Landryová, L.

Kývala, H. *Řízení provozu na modelovém kolejišti*, Ostrava: katedra automatizační techniky a řízení, VŠB-TU Ostrava, 2008. 79 s. Diplomová práce, vedoucí Pavlas, R.

Hynar, M. *Elektrický pohon v distribuovaném systému řízení*. Ostrava: katedra Automatizační techniky a řízení, Fakulta strojní VŠB-Technická univerzita Ostrava, 2008, s. Diplomová práce, vedoucí Formánek, I.

Stránky společnosti Pantek (CS) s.r.o. [online]. 2009. Dostupné z www:

<<http://www.pantek.cz/>>

Automatizace, 9/2007. *Wonderware InTouch 10.0 a Wonderware System Platform 3.0*.

[automatizace\_2007\_9(InTouch10).pdf]. Dostupné z www:

<[http://www.pantek.cz/pdf/press/casopisy/2007/automatizace/automatizace\\_2007\\_9.pdf](http://www.pantek.cz/pdf/press/casopisy/2007/automatizace/automatizace_2007_9.pdf)>

Wonderware 2007. *Wonderware InTouch 10.0*. [intouch\_dtsh.pdf]

Pantek (CS) s.r.o. 2007. *InTouch 10.0 Co je nového?*. [intouch\_noveho.pdf]

Invensys Systems, Inc. *Wonderware OPCLink User's Guide*. 2004.

[IOSrv\_OPCLINK8.pdf]

Invensys Systems, Inc. *Wonderware FactorySuite Gateway User's Guide*. 2004.

[FSGateway.pdf]

Invensys Systems, Inc. *Wonderware InTouch 10.0, Uživatelský manuál*. 2006.

[InTouch\_10\_CZ.pdf]

Invensys Systems, Inc. *InTouch HMI a ArchestrA, Průvodce integrace*. 2007.

[InTouch\_HMI.pdf]

Invensys Systems, Inc. *Tvorba a správa ArchestrA grafiky, Uživatelský manuál*. 2008.

[InTouch\_tvorba.pdf]



Jarka, J. *Wonderware Application Server 3.0 – Co je nového*. Pantek (CS) s.r.o., 2007.  
[ias\_noveho.pdf]

Wonderware, Pantek (CS) s.r.o. *Architektura Archestra – Všechny systémy ve vašem podniku pracují v souladu*. [archestra\_pozice.pdf]. Dostupné z www:  
<[http://www.pantek.cz/pdf/produkty/ias/archestra/archestra\\_pozice.pdf](http://www.pantek.cz/pdf/produkty/ias/archestra/archestra_pozice.pdf) >